To our customers,

Old Company Name in Catalogs and Other Documents

On April 1st, 2010, NEC Electronics Corporation merged with Renesas Technology Corporation, and Renesas Electronics Corporation took over all the business of both companies. Therefore, although the old company name remains in this document, it is a valid Renesas Electronics document. We appreciate your understanding.

Renesas Electronics website: http://www.renesas.com

April 1st, 2010
Renesas Electronics Corporation

Issued by: Renesas Electronics Corporation (http://www.renesas.com)

Send any inquiries to http://www.renesas.com/inquiry.

Notice

1. All information included in this document is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas Electronics products listed herein, please confirm the latest product information with a Renesas Electronics sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas Electronics such as that disclosed through our website.
2. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
3. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part.
4. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
5. When exporting the products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations. You should not use Renesas Electronics products or the technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations.
6. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
7. Renesas Electronics products are classified according to the following three quality grades: "Standard", "High Quality", and "Specific". The recommended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below. You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application categorized as "Specific" without the prior written consent of Renesas Electronics. Further, you may not use any Renesas Electronics product for any application for which it is not intended without the prior written consent of Renesas Electronics. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for an application categorized as "Specific" or for which the product is not intended where you have failed to obtain the prior written consent of Renesas Electronics. The quality grade of each Renesas Electronics product is "Standard" unless otherwise expressly specified in a Renesas Electronics data sheets or data books, etc.

   "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and industrial robots.

   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; safety equipment; and medical equipment not specifically designed for life support.

   "Specific": Aircraft; aerospace equipment; submersible repeaters; nuclear reactor control systems; medical equipment or systems for life support (e.g. artificial life support devices or systems), surgical implantations, or healthcare intervention (e.g. excision, etc.), and any other applications or purposes that pose a direct threat to human life.
8. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

(Note 1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.

(Note 2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

User's Manual

32

SH7709S Group

Hardware Manual

Renesas 32-Bit RISC Microcomputer
SuperH™ RISC engine Family/
SH7700 Series

Renesas Electronics
www.renesas.com

Rev.5.00 2003.09

Renesas 32-Bit RISC Microcomputer
SuperH™ RISC engine Family/SH7700 S

SH7709S Group

Hardware Manual

RENESAS

REJ09B00

Notes regarding these materials

1. These materials are intended as a reference to assist our customers in the selection o
   Renesas Technology Corp. product best suited to the customer's application; they d
   convey any license under any intellectual property rights, or any other rights, belon
   Renesas Technology Corp. or a third party.
2. Renesas Technology Corp. assumes no responsibility for any damage, or infringeme
   third-party's rights, originating in the use of any product data, diagrams, charts, prog
   algorithms, or circuit application examples contained in these materials.
3. All information contained in these materials, including product data, diagrams, char
   programs and algorithms represents information on products at the time of publicati
   materials, and are subject to change by Renesas Technology Corp. without notice d
   product improvements or other reasons. It is therefore recommended that customer
   Renesas Technology Corp. or an authorized Renesas Technology Corp. product dis
   the latest product information before purchasing a product listed herein.
   The information described here may contain technical inaccuracies or typographical
   Renesas Technology Corp. assumes no responsibility for any damage, liability, or o
   rising from these inaccuracies or errors.
   Please also pay attention to information published by Renesas Technology Corp. by
   means, including the Renesas Technology Corp. Semiconductor home page
   (http://www.renesas.com).
4. When using any or all of the information contained in these materials, including pro
   diagrams, charts, programs, and algorithms, please be sure to evaluate all informati
   system before making a final decision on the applicability of the information and pr
   Renesas Technology Corp. assumes no responsibility for any damage, liability or ot
   resulting from the information contained herein.
5. Renesas Technology Corp. semiconductors are not designed or manufactured for us
   device or system that is used under circumstances in which human life is potentially
   Please contact Renesas Technology Corp. or an authorized Renesas Technology Co
   distributor when considering the use of a product contained herein for any specific p
   such as apparatus or systems for transportation, vehicular, medical, aerospace, nucl
   undersea repeater use.
6. The prior written approval of Renesas Technology Corp. is necessary to reprint or r
   whole or in part these materials.
7. If these products or technologies are subject to the Japanese export control restrictio
   must be exported under a license from the Japanese government and cannot be impo
   country other than the approved destination.
   Any diversion or reexport contrary to the export control laws and regulations of Jap
   the country of destination is prohibited.
8. Please contact Renesas Technology Corp. for further details on these materials or th
   contained therein.

2. Treatment of Unused Input Pins

Note: Fix all unused input pins to high or low level.
Generally, the input pins of CMOS products are high-impedance input pins. If
are in their open states, intermediate levels are induced by noise in the vicinity
through current flows internally, and a malfunction may occur.

3. Processing before Initialization

Note: When power is first supplied, the product's state is undefined.
The states of internal circuits are undefined until full power is supplied throug
chip and a low level is input on the reset pin. During the period where the stat
undefined, the register settings and the output state of each pin are also undefi
your system so that it does not malfunction because of processing while it is i
undefined state. For those products which have a reset function, reset the LSI
after the power supply has been turned on.

4. Prohibition of Access to Undefined or Reserved Addresses

Note: Access to undefined or reserved addresses is prohibited.
The undefined or reserved addresses may be used to expand functions, or test
may have been be allocated to these addresses. Do not access these registers; t
operation is not guaranteed if they are accessed.

5. Overview
6. Description of Functional Modules
   - CPU and System-Control Modules
   - On-Chip Peripheral Modules

     The configuration of the functional description of each module differs according module. However, the generic style includes the following items:

   i) Feature
   ii) Input/Output Pin
   iii) Register Description
   iv) Operation
   v) Usage Note

When designing an application system that includes this LSI, take notes into account. E includes notes in relation to the descriptions given, and usage notes are given, as requir final part of each section.

7. List of Registers
8. Electrical Characteristics
9. Appendix
10. Main Revisions and Additions in this Edition (only for revised versions)

The list of revisions is a summary of points that have been revised or added to earlier v This does not include all of the revised contents. For details, see the actual locations in manual.

11. Index

This LSI can be used as a microcomputer for devices that require both high speed and consumption.

Target Readers: This manual is designed for use by people who design application s the SH7709S.
To use this manual, basic knowledge of electric circuits, logic circuits and microcomp required.

Purpose: This manual provides the information of the hardware functions and electric characteristics of the SH7709S.
The SH3, SH-3E, SH3-DSP Programming Manual contains detailed information of e instructions. Please read the Programming Manual together with this manual.

How to Use the Book:

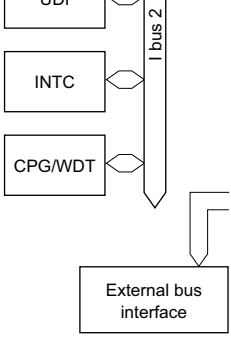
- To understand general functions
  - — Read the manual from the beginning.
    The manual explains the CPU, system control functions, peripheral functions a characteristics in that order.
- To understanding CPU functions
  - — Refer to the separate SH3, SH-3E, SH3-DSP Programming Manual.

Explanatory Note: Bit sequence: upper bit at left, and lower bit at right

List of Related Documents: The latest documents are available on our Web site. Ple sure that you have the latest version.
(http://www.renesas.com/eng/)

- User manuals for SH7709S

Name of Document	Document
SH7709S Group Hardware Manual	This manual
SH3, SH-3E, SH3-DSP Programming Manual	ADE-602-15

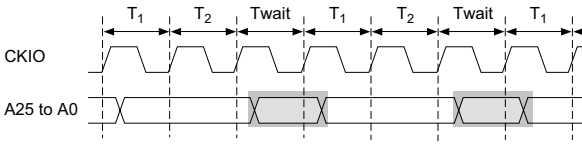


ASERAM deleted from legend

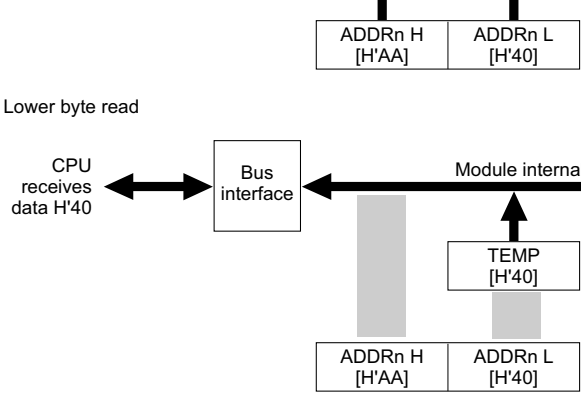
2.5.1 Processor States	53	Description amended In the power-on reset state, the internal states of the on-chip supporting module registers are initialized. In reset state, the internal states of the CPU and registe supporting modules other than the bus state controlle initialized. the register configurations in the relevant sections for details.
5.4 Memory-Mapped Cache 5.4.1 Address Array	113	Description amended This operation is used to invalidate the address speci cache. Write back will take place when the U bit of th received a hit is 1. Note that, when a 0 is written to th should always be written to the U bit of the same entr

		``; R0 = H'0000 0000 LRU = H'000, U = 0, V = 0 ; R1 = H'F000 1080, Way = 1, Entry = H'08, A ; MOV.L R0, @R1```
		To invalidate all entries and ways, write 0 to the following a
		Addresses F000 0000 F000 0010 F000 0020 : F000 3FF0
		This involves a total of 1,024 writes.
		The above operation should be performed using a non-cache
		(2) Invalidating a Specific Address
		Newly added
		(3) Reading Data from a Specific Entry
		Description amended
		``; R0 = H'F100 004C; Data array access, Entry ; Way = 0, Longword address = 3 ; MOV.L R0, @R1 ; Longword 3 is read.```
6.2.6 Interrupt Exception Handling and Priority Table 6.4 Interrupt Exception Handling Sources and Priority (IRQ Mode)	127	IPR (bit numbers) for SCI amended (Before)IPRB(3-0) → (After)IPRB(7-4)
6.3.6 Interrupt Request Register 0 (IRR0)	138	Description amended When clearing an IRQ5R–IRQ0R bit to 0, read the bit v to 1, and then write 0. In this case, 0 should be written bits to be cleared and 1 to the other bits. The contents to which 1 is written do not change.
8.2.1 Standby Control Register (STBCR)	184	Description added Bit 1—Module Standby 1 (MSTP1) Before switching the RTC to module standby, access a among the registers RTC, SCI, and TMU.

9.3 Clock Operating Modes Table 9.4 Available Combinations of Clock Mode and FRQCR Values	210	2. under cautions amended The peripheral clock frequency should not be set high frequency of the CKIO pin, higher than 33.34 MHz.
9.5.1 Changing the Multiplication Rate	213	Description added 5.Supply of the clock that has been set begins at WD overflow, and the processor begins operating again stops after it overflows. When the following three conditions are all met, FRQ not be changed while a DMAC transfer is in progress • Bits IFC2 to IFC0 are changed. • STC2 to STC0 are not changed. • The clock ratio of Iϕ (on-chip clock) to Bϕ (bus cloc change is other than 1:1.
9.8.2 Changing the Frequency	218, 219	Description added 5.The counter stops at a value of H'00 or H'01. The st depends on the clock ratio. When the following three conditions are all met, FRQ not be changed while a DMAC transfer is in progress • Bits IFC2 to IFC0 are changed. • STC2 to STC0 are not changed. • The clock ratio of Iϕ (on-chip clock) to Bϕ (bus cloc change is other than 1:1.
10.1.1 Features	223	Refresh function description deleted
10.2.5 Individual Memory Control Register (MCR)	246	Description added **Bit 7—Synchronous DRAM Bank Active (RASD):** S whether synchronous DRAM is used in bank active m precharge mode. Set auto-precharge mode when are are both designated as synchronous DRAM space. The bank active mode should not be used unless the for all areas is 32 bits.

		The bank active mode should not be used unless the b for all areas is 32 bits.
10.3.6 PCMCIA Interface Figure 10.32 Basic Timing for PCMCIA Memory Card Interface	310	Figure amended D15 to D0 (Write)
10.3.7 Waits between Access Cycles Figure 10.40 Waits between Access Cycles	320	Figure amended 
10.3.10 $\overline{\text{MCS[0]}}$ to $\overline{\text{MCS[7]}}$ Pin Control	323	Description amended This enables 32-, 64-, 128-, or 256-Mbit memory to be to area 0 or area 2. However, only CS2/0 = 0 (area 0) s used for MCSCR0. Table 10.15 shows MCSCR0–MCS settings and $\overline{\text{MCS[0]}}$–$\overline{\text{MCS[7]}}$ assertion conditions.
11.6 Usage Notes	387	Description added 13. DMAC transfers should not be performed in the sl under conditions other than when the clock ratio o chip clock) to Bϕ (bus clock) is 1:1. 14. When the following three conditions are all met, th frequency control register (FRQCR) should not be while a DMAC transfer is in progress. • Bits IFC2 to IFC0 are changed. • STC2 to STC0 in FRQCR are not changed. • The clock ratio of Iϕ (on-chip clock) to Bϕ (bus c the change is other than 1:1.
13.4.3 Precautions when Using RTC Module Standby	426	Newly added

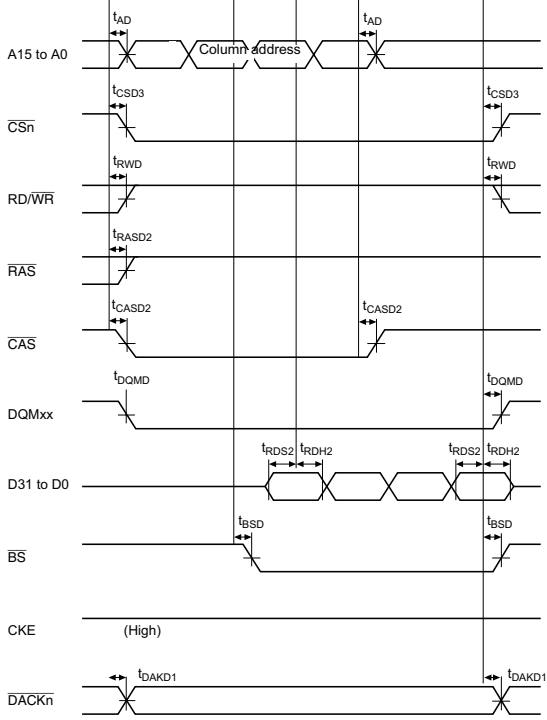
		When the RDF flag in SCSSR is set to 1, an RXI inter is generated. The DMAC can be activated and data t performed when the RDF flag in SCSSR is set to 1. W receive data less than the receive trigger number is re receive data register (SCFRDR) by the DMAC, 1 is re RDF flag, after which 0 is written to it to clear it.
16.5 Usage Notes	551	Description amended 1. SCFTDR Writing and TDFE Flag: However, if the number of data bytes written to SCFT to or less than the transmit trigger number, the TDFE set to 1 again even after having been cleared to 0. TD should therefore be carried out after data exceeding t transmit trigger number has been written to SCFTDR. 2. SCFRDR Reading and RDF Flag: However, if the number of data bytes in SCFRDR exc trigger number, the RDF flag will be set to 1 again eve having been cleared to 0. RDF should therefore be cl after being read as 1 after all the receive data has be
19.13.2 SC Port Data Register (SCPDR)	610	Title Amended



23.1 Absolute Maximum Ratings Table 23.1 Absolute Maximum Ratings	657	Caution added 2.Until voltage is applied to all power supplies, a low le at the $\overline{\text{RESETP}}$ pin, and CKIO has operated for a maxi clock cycles, internal circuits remain unsettled, and so are also undefined. The system design must ensure th undefined states do not cause erroneous system opera Note that the $\overline{\text{RESETP}}$ pin cannot receive a low level s a low level signal is being input to the CA pin.
23.2 DC Characteristics Table 23.2 DC Characteristics	659, 662	Test conditions for in sleep mode amended

Item	Symbol	Min	Typ	Max	Unit	Test Cor
Sleep mode*1	Icc	—	15	30		*1: When other exte cycle oth refresh cy
	IccQ	—	10	20		Vcc = 1.9 VccQ = 3 Bϕ = 33M

Note * added
* If the IRL and IRLS interrupts are used, the minimu


tAD
tAD
A15 to A0
Column address
tCSD3
tCSD3
CSn
tRWD
tRWD
RD/WR
tRASD2
RAS
tCASD2
tCASD2
CAS
tDQMD
tDQMD
DQMxx
tRDS2
tRDH2
tRDS2
tRDH2
D31 to D0
tBSD
tBSD
BS
CKE
(High)
tDAKD1
tDAKD1
DACKn

		V_{CC}–PLL1 V_{CC}–PLL2	145 150	F16, E17	Power supply	PLL power su (2.0/1.9/1.8/1.
		V_{CC}	29, 81, 134, 154, 175	L3, L4, U11, T11, J17, J16, E18, C19, C12, D12	Power supply	Internal powe (2.0/1.9/1.8/1.

A.3 Treatment of Unused Pins	724	"When RTC is not used" and "When PLL2 is not used" (Before) (1.9/1.8V) →(After) (2.0/1.9/1.8/1.7V)
A.4 Pin States in Access to Each Address Space Table A.3 Pin States (Ordinary Memory/Little Endian) Table A.4 Pin States (Ordinary Memory/Big Endian) Table A.5 Pin States (Burst ROM/Little Endian) Table A.6 Pin States (Burst ROM/Big Endian) Table A.9 Pin States (PCMCIA/Little Endian) Table A.10 Pin States (PCMCIA/Big Endian)	726 to 738	Note 2 amended Note: 2. Unused data pins should be switched to the p function, or pulled up.

interface. This LSI includes data protection, virtual memory, and other functions prov
incorporating an MMU into a SuperH Series microprocessor (SH-1 or SH-2).

High-speed data transfers can be performed by an on-chip direct memory access contr
(DMAC) and an external memory access support function enables direct connection to
types of memory. The SH7709S microprocessor also supports an infrared communica
function, an A/D converter, and a D/A converter.

A powerful built-in power management function keeps power consumption low, even
speed operation. This LSI can run at six times the frequency of the system bus operati
making it optimum for electrical devices such as PDAs that require both high speed an
power.

The features of this LSI is listed in table 1.1. The specifications are shown in table 1.2

Note: SuperH is a trademark of Renesas Technology, Corp.

	— Eight 32-bit control registers — Four 32-bit system registers • RISC-type instruction set — Instruction length: 16-bit fixed length for improved code efficie — Load-store architecture — Delayed branch instructions — Instruction set based on C language • Instruction execution time: one instruction/cycle for basic instructi • Logical address space: 4 Gbytes • Space identifier ASID: 8 bits, 256 logical address space • Five-stage pipeline
Clock pulse generator (CPG)	• Clock mode: An input clock can be selected from the external inp or CKIO) or crystal oscillator. • Three types of clocks generated: — CPU clock: 1–24 times the input clock, maximum 200 MHz — Bus clock: 1–4 times the input clock, maximum 66.67 MHz — Peripheral clock: 1/4–4 times the input clock, maximum 33.34 • Power-down modes: — Sleep mode — Standby mode — Module standby mode • One-channel watchdog timer
Memory management unit (MMU)	• 4 Gbytes of address space, 256 address spaces (ASID 8 bits) • Page unit sharing • Supports multiple page sizes: 1, 4 kbytes • 128-entry, 4-way set associative TLB • Supports software selection of replacement method and random- algorithms

User break controller (UBC)	• 2 break channels • Addresses, data values, type of access, and data size can all be conditions • Supports a sequential break function
Bus state controller (BSC)	• Physical address space divided into six areas (area 0, areas 2 t maximum of 64 Mbytes, with the following features settable for e — Bus size (8, 16, or 32 bits) — Number of wait cycles (also supports a hardware wait functi — Setting the type of space enables direct connection to SRAM Synchronous DRAM, and burst ROM — Supports PCMCIA interface (2 channels) — Outputs chip select signal (CS0, CS2–CS6) for correspondi • Synchronous DRAM refresh function — Programmable refresh interval — Support self-refresh mode • Synchronous DRAM burst access function • Usable as either big or little endian machine
User-debugging Interface (UDI)	• E10A emulator support • JTAG-compliant • Realtime branch address trace • 1-kB on-chip RAM for fast emulation program execution
Timer (TMU)	• 3-channel auto-reload-type 32-bit timer • Input capture function • 6 types of counter input clocks can be selected • Maximum resolution: 2 MHz
Realtime clock (RTC)	• Built-in clock, calendar functions, and alarm functions • On-chip 32-kHz crystal oscillator circuit with a maximum resoluti cycle) of 1/256 second

cation interface 2 (SCI2/SCIF)	• DMA can be transferred • Hardware flow control
Direct memory access controller (DMAC)	• 4 channels • Burst mode and cycle-steal mode • Data transfer size: 8-/16-/32-bit and 16-byte
I/O port	• Twelve 8-bit ports
A/D converter (ADC)	• 10 bits ± 4 LSB, 8 channels • Conversion time: 16 µs • Input range: 0–AVcc (max. 3.6 V)
D/A converter (DAC)	• 8 bits ± 4 LSB, 2 channels • Conversion time: 10 µs • Output range: 0–AVcc (max. 3.6 V)

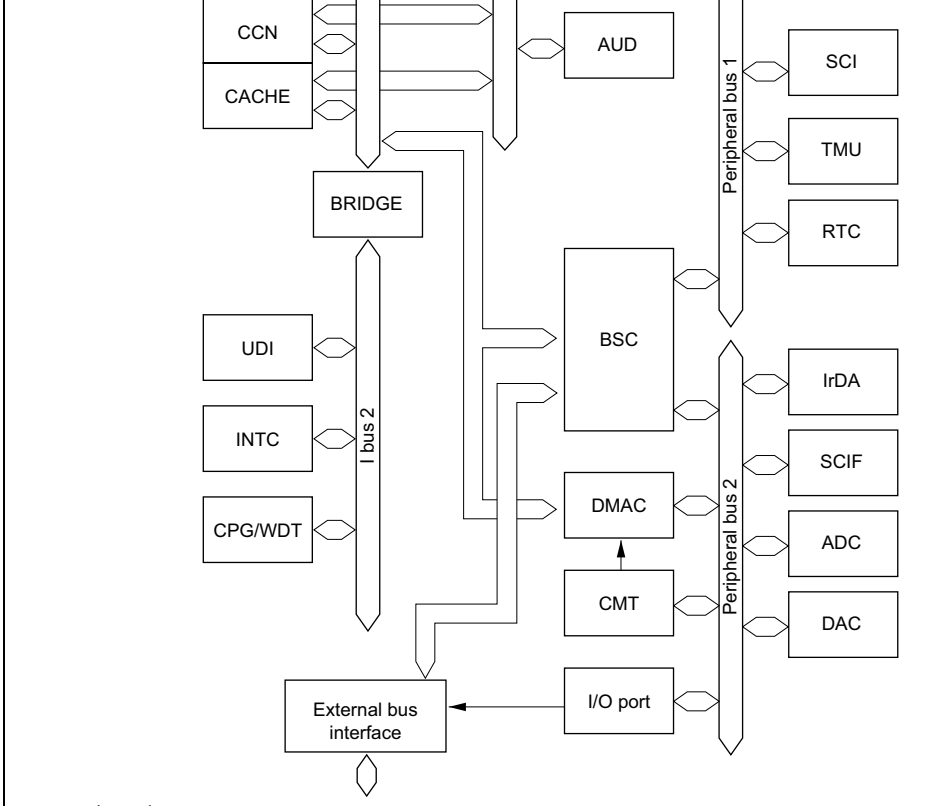
Product lineup

	Power Supply Voltage		Operating		
Abbr.	**I/O**	**Internal**	**Frequency**	**Model Name**	**Pac**
SH7709S	3.3±0.3 V	2.0±0.15 V*	200 MHz	HD6417709SHF200B	208 HQ
		1.9±0.15 V	167 MHz	HD6417709SF167B	208 LQ
				HD6417709SBP167B	240 (BP
		1.8+0.25 V 1.8–0.15 V	133 MHz	HD6417709SF133B	208 LQ
				HD6417709SBP133B	240 (BP
		1.7+0.25 V 1.7–0.15 V	100 MHz	HD6417709SF100B	208 LQ
				HD6417709SBP100B	240 (BP

Note: * 2.0 (+0.15, –0.1) V when an IRL or IRLS interrupt is used.

	model); external frequency: maximum 66.67 MHz
Process	• 0.25-µm CMOS/5-layer metal

Note: * 2.0 (+0.15, –0.1) V when an IRL or IRLS interrupt is used.



Legend:

ADC:	A/D converter	INTC:	Interrupt controller
AUD:	Advanced user debugger	IrDA:	Serial communicatiion interface (with IrDA)
BSC:	Bus state controller	MMU:	Memory management unit
CACHE:	Cache memory	RTC:	Realtime clock
CCN:	Cache memory controller	SCI:	Serial communication interface (with smart card int
CMT:	Compare match timer	SCIF:	Serial communication interface (with FIFO)
CPG/WDT:	Clock pulse generator/watchdog timer	TLB:	Address translation buffer
CPU:	Central processing unit	TMU:	Timer unit
DAC:	D/A converter	UBC:	User break controller
DMAC:	Direct memory access controller		
UDI:	User debugging interface		

Figure 1.1 Block Diagram

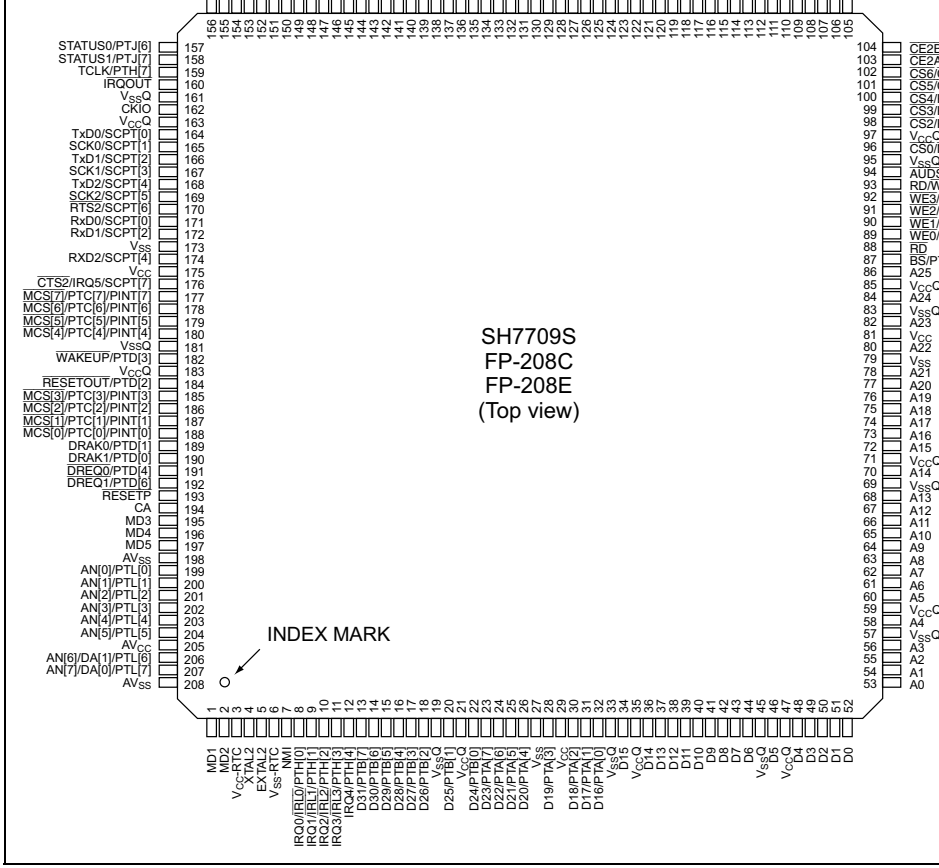
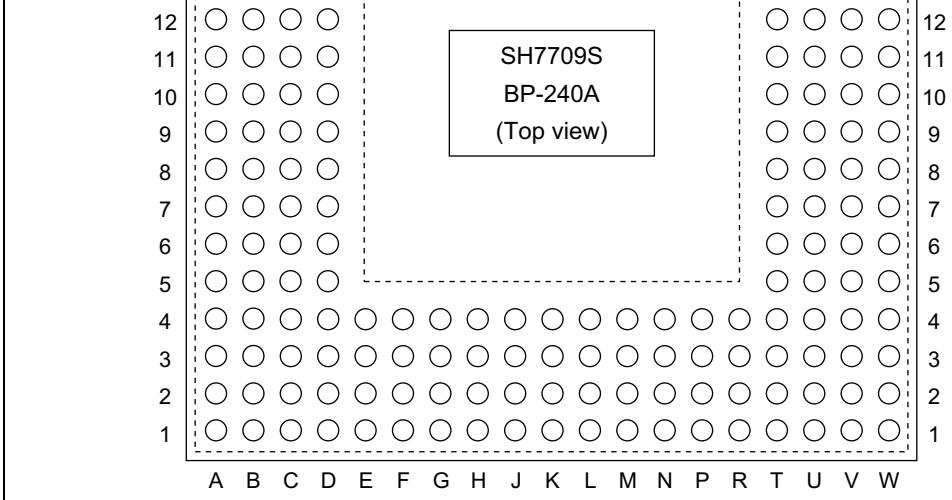


Figure 1.2 Pin Assignment (FP-208C, FP-208E)



Note: The pin area enclosed in broken lines is an inner view.

Figure 1.3 Pin Assignment (BP-240A)

3	E2	Vcc-RTC*1	—	RTC power supply (*
4	D1	XTAL2	O	On-chip RTC crystal
5	D3	EXTAL2	I	On-chip RTC crystal pin*6
6	E1	Vss-RTC*1	—	RTC power supply (0
7	C3	NMI	I	Nonmaskable interrup
8	E3	IRQ0/$\overline{\text{IRL0}}$/PTH[0]	I	External interrupt requ port H
9	E4	IRQ1/$\overline{\text{IRL1}}$/PTH[1]	I	External interrupt requ port H
10	F1	IRQ2/$\overline{\text{IRL2}}$/PTH[2]	I	External interrupt requ port H
11	F2	IRQ3/$\overline{\text{IRL3}}$/PTH[3]	I	External interrupt requ port H
12	F3	IRQ4/PTH[4]	I	External interrupt requ port H
13	F4	D31/PTB[7]	I/O	Data bus / input/outpu
14	G1	D30/PTB[6]	I/O	Data bus / input/outpu
15	G2	D29/PTB[5]	I/O	Data bus / input/outpu
16	G3	D28/PTB[4]	I/O	Data bus / input/outpu
17	G4	D27/PTB[3]	I/O	Data bus / input/outpu
18	H1	D26/PTB[2]	I/O	Data bus / input/outpu
19	H2	VssQ	—	Input/output power su
20	H3	D25/PTB[1]	I/O	Data bus / input/outpu
21	H4	VccQ	—	Input/output power su
22	J1	D24/PTB[0]	I/O	Data bus / input/outpu
23	J2	D23/PTA[7]	I/O	Data bus / input/outpu
24	J4	D22/PTA[6]	I/O	Data bus / input/outpu
25	J3	D21/PTA[5]	I/O	Data bus / input/outpu
26	K2	D20/PTA[4]	I/O	Data bus / input/outpu

—	L4	Vcc	—	Power supply (*3)
30	L2	D18/PTA[2]	I/O	Data bus / input/output
31	L1	D17/PTA[1]	I/O	Data bus / input/output
32	M4	D16/PTA[0]	I/O	Data bus / input/output
33	M3	VssQ	—	Input/output power sup
34	M2	D15	I/O	Data bus
35	M1	VccQ	—	Input/output power sup
36	N4	D14	I/O	Data bus
37	N3	D13	I/O	Data bus
38	N2	D12	I/O	Data bus
39	N1	D11	I/O	Data bus
40	P4	D10	I/O	Data bus
41	P3	D9	I/O	Data bus
42	P2	D8	I/O	Data bus
43	P1	D7	I/O	Data bus
44	R4	D6	I/O	Data bus
45	R3	VssQ	—	Input/output power sup
46	T4	D5	I/O	Data bus
47	R1	VccQ	—	Input/output power sup
48	T3	D4	I/O	Data bus
49	T1	D3	I/O	Data bus
50	R2	D2	I/O	Data bus
51	U2	D1	I/O	Data bus
52	T2	D0	I/O	Data bus
53	V4	A0	O	Address bus
54	V3	A1	O	Address bus
55	V5	A2	O	Address bus
56	W4	A3	O	Address bus

62	W6	A7	O	Address bus
63	V6	A8	O	Address bus
64	U6	A9	O	Address bus
65	T6	A10	O	Address bus
66	W7	A11	O	Address bus
67	V7	A12	O	Address bus
68	U7	A13	O	Address bus
69	T7	VssQ	—	Input/output power su
70	W8	A14	O	Address bus
71	V8	VccQ	—	Input/output power su
72	U8	A15	O	Address bus
73	T8	A16	O	Address bus
74	W9	A17	O	Address bus
75	V9	A18	O	Address bus
76	T9	A19	O	Address bus
77	U9	A20	O	Address bus
78	V10	A21	O	Address bus
79	U10	Vss	—	Power supply (0 V)
—	T10	Vss	O	Power supply (0 V)
80	W10	A22	O	Address bus
81	U11	Vcc	—	Power supply (*3)
—	T11	Vcc	—	Power supply (*3)
82	V11	A23	O	Address bus
83	W11	VssQ	—	Input/output power su
84	T12	A24	O	Address bus
85	U12	VccQ	—	Input/output power su
86	V12	A25	O	Address bus

90	V13	WE1/DQMLL/WE	O	D15–D8 select signal / (SDRAM)
91	W13	WE2/DQMUL/ICIORD/ PTK[6]	O / I/O	D23–D16 select signal (SDRAM) / PCMCIA I/O input/output port K
92	T14	WE3/DQMUU/ICIOWR/ PTK[7]	O / I/O	D31–D24 select signal (SDRAM) / PCMCIA I/O input/output port K
93	U14	RD/WR	O	Read/write
94	V14	AUDSYNC/PTE[7]	O / I/O	AUD synchronous / inp port E
95	W14	VssQ	—	Input/output power sup
96	T15	CS0/MCS[0]	O	Chip select 0/mask RO select 0
97	U15	VccQ	—	Input/output power sup
98	T16	CS2/PTK[0]	O / I/O	Chip select 2 / input/ou
99	W15	CS3/PTK[1]	O / I/O	Chip select 3 / input/ou
100	U16	CS4/PTK[2]	O / I/O	Chip select 4 / input/ou
101	W16	CS5/CE1A/PTK[3]	O / I/O	Chip select 5/CE1 (are PCMCIA) / input/outpu
102	V15	CS6/CE1B	O	Chip select 6/CE1 (are PCMCIA)
103	V17	CE2A/PTE[4]	O / I/O	CE2 (area 5 PCMCIA) input/output port E
104	V16	CE2B/PTE[5]	O / I/O	CE2 (area 6 PCMCIA) input/output port E
105	T18	CKE/PTK[5]	O / I/O	CK enable (SDRAM) / port K

109	T19	VSSQ	—	Input/output power su
110	T17	$\overline{\text{CASU}}$/PTJ[3]	O / I/O	Lower 32 Mbytes add (SDRAM) CAS / input
111	R19	VccQ	—	Input/output power su
112	U17	PTJ[4]	I/O	Input/output port J
113	R17	PTJ[5]	I/O	Input/output port J
114	R16	DACK0/PTD[5]	O / I/O	DMA acknowledge 0 / port D
115	P19	DACK1/PTD[7]	O / I/O	DMA acknowledge 1 / port D
116	P18	PTE[6]	I/O	Input/output port E
117	P17	PTE[3]	I/O	Input/output port E
118	P16	$\overline{\text{RAS3U}}$/PTE[2]	O / I/O	Upper 32 Mbytes add (SDRAM) RAS / input E
119	N19	PTE[1]	I/O	Input/output port E
120	N18	TDO/PTE[0]	O / I/O	Test data output / inp port E
121	N17	$\overline{\text{BACK}}$	O	Bus acknowledge
122	N16	$\overline{\text{BREQ}}$	I	Bus request
123	M19	$\overline{\text{WAIT}}$	I	Hardware wait reques
124	M18	$\overline{\text{RESETM}}$	I	Manual reset request
125	M17	$\overline{\text{ADTRG}}$/PTH[5]	I	Analog trigger / input
126	M16	$\overline{\text{IOIS16}}$/PTG[7]	I	IOIS16 (PCMCIA) / in
127	L19	$\overline{\text{ASEMD0}}$/PTG[6]	I	ASE mode[*4] / input p
128	L18	$\overline{\text{ASEBRKAK}}$/PTG[5]	O/I	ASE break acknowled port G
129	L16	PTG[4]/CK102	I	Input port G / clock ou

134	J17	Vcc	—	Power supply (*3)
—	J16	Vcc	—	Power supply (*3)
135	J18	AUDATA[0]/PTG[0]	I/O / I	AUD data / input port G
136	J19	$\overline{\text{TRST}}$/PTF[7]/PINT[15]	I	Test reset / input port F interrupt
137	H16	TMS/PTF[6]/PINT[14]	I	Test mode switch / inpu port interrupt
138	H17	TDI/PTF[5]/PINT[13]	I	Test data input / input p interrupt
139	H18	TCK/PTF[4]/PNT[12]	I	Test clock / input port F interrupt
140	H19	$\overline{\text{IRLS3}}$/PTF[3]/ PINT[11]	I	External interrupt requ port F / port interrupt
141	G16	$\overline{\text{IRLS2}}$/PTF[2]/ PINT[10]	I	External interrupt requ port F / port interrupt
142	G17	$\overline{\text{IRLS1}}$/PTF[1]/PINT[9]	I	External interrupt requ port F / port interrupt
143	G18	$\overline{\text{IRLS0}}$/PTF[0]/PINT[8]	I	External interrupt requ port F / port interrupt
144	G19	MD0	I	Clock mode setting
145	F16	Vcc-PLL1*2	—	PLL1 power supply (*3
146	F17	CAP1	—	PLL1 external capacita
147	F18	Vss-PLL1*2	—	PLL1 power supply (0 V
148	F19	Vss-PLL2*2	—	PLL2 power supply (0 V
149	E16	CAP2	—	PLL2 external capacita
150	E17	Vcc-PLL2*2	—	PLL2 power supply (*3
151	D16	AUDCK/PTH[6]	I	AUD clock / input port H
152	E19	Vss	—	Power supply (0 V)
153	D17	Vss	—	Power supply (0 V)

				pin
157	B16	STATUS0/PTJ[6]	O / I/O	Processor status / inp port J
158	B17	STATUS1/PTJ[7]	O / I/O	Processor status / inp port J
159	B15	TCLK/PTH[7]	I/O	TMU or RTC clock inp input/output port H
160	A16	$\overline{\text{IRQOUT}}$	O	Interrupt request notif
161	C16	VssQ	—	Input/output power su
162	A15	CKIO	I/O	System clock input/ou
163	C17	VccQ	—	Power supply (3.3 V)
164	C15	TxD0/SCPT[0]	O	Transmit data 0 / SCI
165	D15	SCK0/SCPT[1]	I/O	Serial clock 0 / SCI in port
166	A14	TxD1/SCPT[2]	O	Transmit data 1 / SCI
167	B14	SCK1/SCPT[3]	I/O	Serial clock 1 / SCI in port
168	C14	TxD2/SCPT[4]	O	Transmit data 2 / SCI
169	D14	SCK2/SCPT[5]	I/O	Serial clock 2 / SCI in port
170	A13	$\overline{\text{RTS2}}$/SCPT[6]	O / I/O	Transmit request 2 / S input/output port
171	B13	RxD0/SCPT[0]	I	Transmit data 0 / SCI
172	C13	RxD1/SCPT[2]	I	Transmit data 1 / SCI
173	D13	Vss	—	Power supply (0 V)
—	A12	Vss	—	Power supply (0 V)
174	B12	RxD2/SCPT[4]	I	Transmit data 2 / SCI
175	C12	Vcc	—	Power supply (*3)
—	D12	Vcc	—	Power supply (*3)

179	C11	$\overline{\text{MCS}}$[5]/PTC[5]/PINT[5]	O / I/O / I	Mask ROM chip select input/output port C / po
180	B10	$\overline{\text{MCS}}$[4]/PTC[4]/PINT[4]	O / I/O / I	Mask ROM chip select input/output port C / po
181	C10	VssQ	—	Input/output power sup
182	D10	$\overline{\text{WAKEUP}}$/PTD[3]	O / I/O	Standby mode interrup notification / input/outp
183	A10	VccQ	—	Input/output power sup
184	C9	$\overline{\text{RESETOUT}}$/PTD[2]	O / I/O	Reset output / input/ou
185	D9	$\overline{\text{MCS}}$[3]/PTC[3]/PINT[3]	O / I/O / I	Mask ROM chip select input/output port C / po
186	B9	$\overline{\text{MCS}}$[2]/PTC[2]/PINT[2]	O / I/O / I	Mask ROM chip select input/output port C / po
187	A9	$\overline{\text{MCS}}$[1]/PTC[1]/PINT[1]	O / I/O / I	Mask ROM chip select input/output port C / po
188	D8	$\overline{\text{MCS}}$[0]/PTC[0]/PINT[0]	O / I/O / I	Mask ROM chip select input/output port C / po
189	C8	DRAK0/PTD[1]	O / I/O	DMA request acknowle input/output port D
190	B8	DRAK1/PTD[0]	O / I/O	DMA request acknowle input/output port D
191	A8	$\overline{\text{DREQ0}}$/PTD[4]	I	DMA request / input po
192	D7	$\overline{\text{DREQ1}}$/PTD[6]	I	DMA request / input po
193	C7	$\overline{\text{RESETP}}$	I	Power-on reset reques
194	B7	CA	I	Chip activate (hardwar request signal)
195	A7	MD3	I	Area 0 bus width settin
196	D6	MD4	I	Area 0 bus width settin
197	C6	MD5	I	Endian setting

203	A5	AN[4]/PTL[4]	I	A/D converter input / i
204	C4	AN[5]/PTL[5]	I	A/D converter input / i
205	A4	AVcc	—	Analog power supply
206	B5	AN[6]/DA[1]/PTL[6]	I	A/D converter input / D/A converter output
207	B3	AN[7]/DA[0]/PTL[7]	I	A/D converter input / D/A converter output
208	B4	AVss	—	Analog power supply

Notes:
1. Must be connected to the power supply even when the RTC is not used.
2. Except in hardware standby mode, all of the power supply pins must be co the system power supply. (Supply power constantly.) In hardware standby must be supplied at least to V_{CC} –RTC and V_{SS} –RTC. If power is not being any of the power supply pins other than V_{CC} –RTC and V_{SS} –RTC, hold the
3. 2.0 V for the 200 MHz model, 1.9 V for the 167 MHz model, 1.8 V for the 13 model, 1.7 V for the 100 MHz model.
4. When this LSI is used on the user system alone, without an emulator and t this pin at high level. When this pin is low or open, $\overline{\text{RESETP}}$ may be maske section 22, User Debugging Interface (UDI)).
5. B2, B1, C1, U1, V1, W1, V2, W2, W3, W17, W18, W19, V18, V19, B19, A1 A17, A3, A2, and A1 are NC pins. Do not connect anything to these pins.
6. If EXTAL2 is not used, pull this pin up to the Vcc-RTC level.

or an interrupt is accepted. There are three kinds of registers—general registers, syster
and control registers—and the registers that can be accessed differ in the two processo

General Registers: There are 16 general registers, designated R0 to R15. General reg
R7 are banked registers which are switched by a processor mode change. In privileged
register bank bit (RB) in the status register (SR) defines which banked register set is a
general registers, and which set is accessed only through the load control register (LDC
control register (STC) instructions.

When the RB bit is 1, the 16 registers comprising BANK1 general registers R0_BANK
R7_BANK1 and non-banked general registers R8–R15 function as the general registe
the 8 registers comprising BANK0 general registers R0_BANK0–R7_BANK0 access
the LDC/STC instructions.

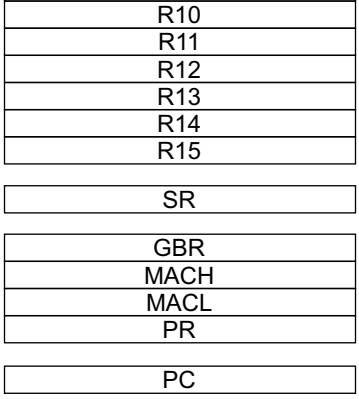
When the RB bit is 0, BANK0 general registers R0_BANK0–R7_BANK0 and nonban
registers R8–R15 function as the general register set, with BANK1 general registers R
R7_BANK1 accessed only by the LDC/STC instructions. In user mode, the 16 registe
comprising bank 0 general registers R0_BANK0–R7_BANK0 and non-banked registe
can be accessed as general registers R0–R15, and bank 1 general registers R0_BANK
R7_BANK1 cannot be accessed.

Control Registers: Control registers comprise the global base register (GBR) and stat
(SR) which can be accessed in both processor modes, and the saved status register (SS
program counter (SPC), and vector base register (VBR) which can only be accessed in
mode. Some bits of the status register (such as the RB bit) can only be accessed in priv
mode.

System Registers: System registers comprise the multiply and accumulate registers
(MACL/MACH), the procedure register (PR), and the program counter (PC). Access t
registers does not depend on the processor mode.

The register configuration in each mode is shown in figures 2.1 and 2.2.

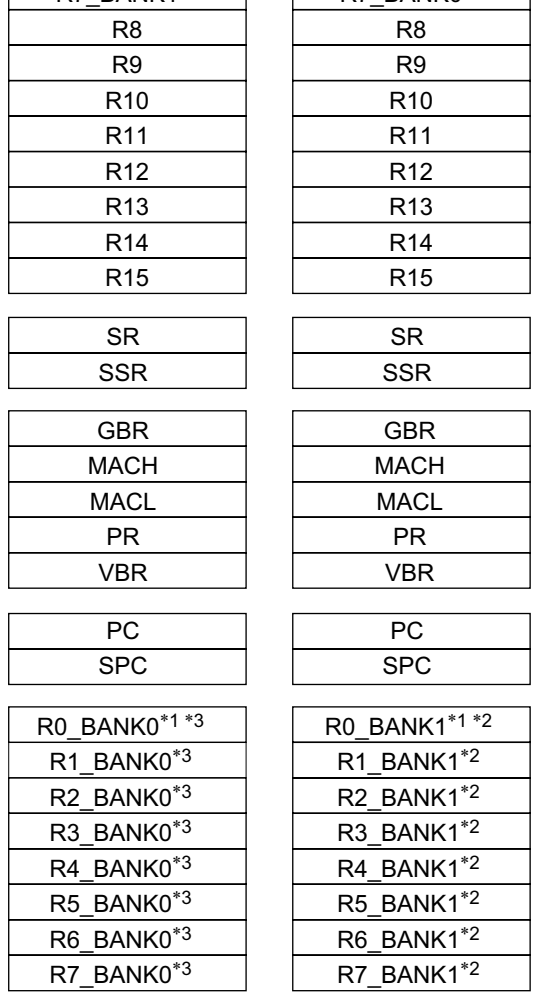
Switching between user mode and privileged mode is controlled by the processor mod
in the status register.



User mode register configuration

Notes: 1. R0 functions as an index register in the indexed register-indirect addre
mode and indexed GBR-indirect addressing mode.
2. Banked register.

Figure 2.1 User Mode Register Configuration



R8
R9
R10
R11
R12
R13
R14
R15

SR
SSR

GBR
MACH
MACL
PR
VBR

PC
SPC

R0_BANK0*1 *3
R1_BANK0*3
R2_BANK0*3
R3_BANK0*3
R4_BANK0*3
R5_BANK0*3
R6_BANK0*3
R7_BANK0*3

a. Privileged mode register configuration (RB = 1)

R8
R9
R10
R11
R12
R13
R14
R15

SR
SSR

GBR
MACH
MACL
PR
VBR

PC
SPC

R0_BANK1*1 *2
R1_BANK1*2
R2_BANK1*2
R3_BANK1*2
R4_BANK1*2
R5_BANK1*2
R6_BANK1*2
R7_BANK1*2

b. Privileged mode register configuration (RB = 0)

Notes: 1. R0 functions as an i
register in the index
register-indirect add
mode and indexed
indirect addressing

2. Banked register
When the RB bit of
register is 1, the reg
be accessed for ger
When the RB bit is
only be accessed w
LDC/STC instructio

3. Banked register
When the RB bit of
register is 0, the reg
be accessed for ger
When the RB bit is
only be accessed w
LDC/STC instructio

Figure 2.2 Privileged Mode Register Configuration

	GBR, SSR, SPC	Undefined
	VBR	H'00000000
System registers	MACH, MACL, PR	Undefined
	PC	H'A0000000

Note: * Register values are initialized at power-on reset or manual reset.

2.1.2 General Registers

There are 16 general registers, designated R0 to R15 (figure 2.3). General registers R0 banked registers, with a different R0–R7 register bank (R0_BANK0–R7_BANK0 or R0_BANK1–R7_BANK1) being accessed according to the processor mode. For details figures 2.1 and 2.2.

The general register configuration is shown in figure 2.3.

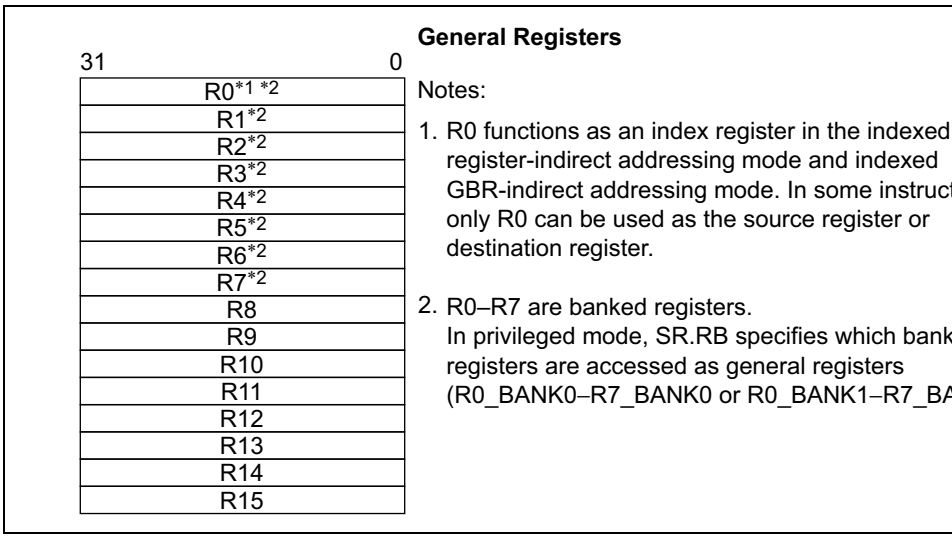


Figure 2.3 General Registers

- Procedure register (PR)
- Program counter (PC)

The system register configuration is shown in figure 2.4.

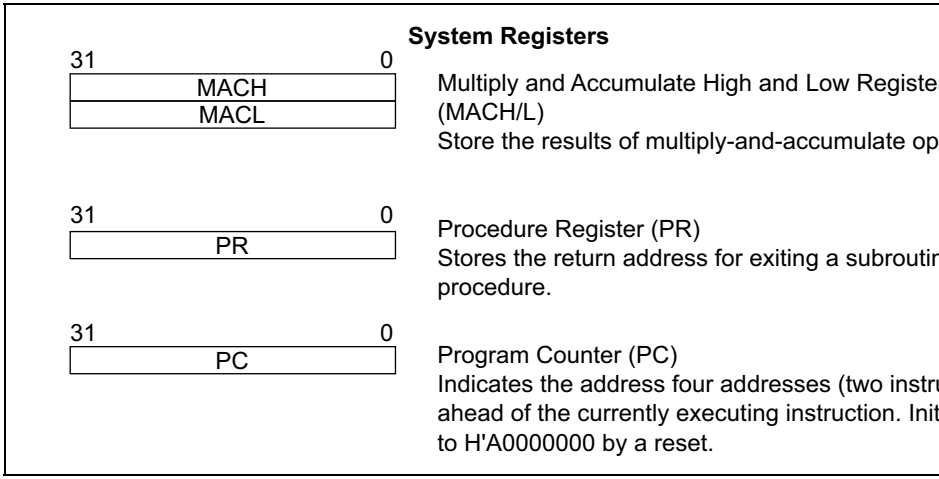


Figure 2.4 System Registers

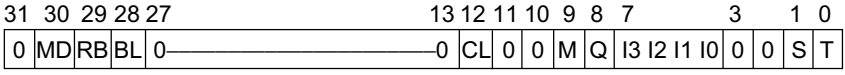
2.1.4 Control Registers

Control registers can be accessed in privileged mode using the LDC and STC instructi
GBR register can also be accessed in user mode. There are five control registers, as fo

- Status register (SR)
- Saved status register (SSR)
- Saved program counter (SPC)
- Global base register (GBR)
- Vector base register (VBR)

addressing mode. The GBR-indirect addressing
is used for on-chip supporting module register a
data transfers and logic operations.
The GBR register can also be accessed in user
Its contents are undefined after a reset.

Vector Base Register (VBR)
Stores base address of exception handling vect
Initialized to H'0000000 by a reset.



S
re
(S

MD: Processor operation mode bit: Indicates the processor operation mode as follows:
MD =1: Privileged mode; MD = 0: User mode
MD is set to 1 on generation of an exception or interrupt , and is initialized to 1 by a

RB: Register bank bit: Determines the bank of general registers R0–R7 used in process
RB = 1: R0_BANK1–R7_BANK1 and R8–R15 are general registers, and R0_BANK
R7_BANK0 can be accessed by LDC/STC instructions.
RB = 0: R0_BANK0–R7_BANK0 and R8–R15 are general registers, and R0_BANK
R7_BANK1 can be accessed by LDC/STC instructions.
RB is set to 1 on generation of an exception or interrupt , and is initialized to 1 by a

BL: Block bit
BL = 1: Exceptions and interrupts are suppressed. See section 4, Exception Handling, for details.
BL = 0: Exceptions and interrupts are accepted.
BL is set to 1 on generation of an exception or interrupt , and is initialized to 1 by a

CL: Cache lock bit
When set to 1, the cache lock function can be used.

M and Q bits: Used by the DIV0S/U and DIV1 instructions.

I3–I0 bits: Interrupt mask bits: 4-bit field indicating the interrupt request mask level.
I3–I0 do not change to the interrupt acceptance level when an interrupt is generated
Initialized to B'1111 by a reset.

S bit: Used by the MAC instruction.

T bit: Used by the MOVT, CMP/cond, TAS, TST, BT, BF, SETT, CLRT, and DT instructio
indicate true (1) or false (0).
Used by the ADDV/C, SUBV/C, DIV0U/S, DIV1, NEGC, SHAR/L, SHLR/L, ROTR/L
ROTCR/L instructions to indicate a carry, borrow, overflow, or underflow.

0 bits: These bits always read 0, and the write value should always be 0.

Note: The M, Q, S, and T bits can be set or cleared by special instructions in user mode.
Their values are undefined after a reset. All other bits can be read or written in privileged m

Figure 2.5 Register Set Overview, Control Registers

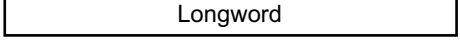


Figure 2.6 Longword

2.2.2 Data Format in Memory

Memory data formats are classified into bytes, words, and longwords. Memory can be 8-bit byte, 16-bit word, or 32-bit longword form. A memory operand less than 32 bits sign-extended before being stored in a register.

A word operand must be accessed starting from a word boundary (even address of a 2 address 2n), and a longword operand starting from a longword boundary (even addres unit: address 4n). An address error will result if this rule is not observed. A byte opera accessed from any address.

Big-endian or little-endian byte order can be selected for the data format. The endian m be set with the MD5 external pin in a power-on reset. Big-endian mode is selected wh pin is low, and little-endian when high. The endian mode cannot be changed dynamica positions are numbered left to right from most-significant to least-significant. Thus, in longword, the leftmost bit, bit 31, is the most significant bit and the rightmost bit, bit 0 significant bit.

The data format in memory is shown in figure 2.7.

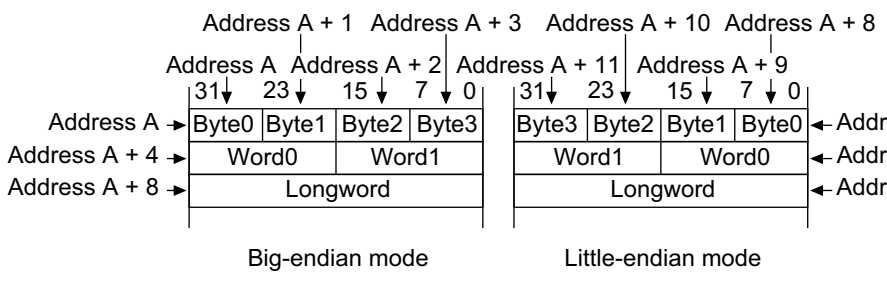


Figure 2.7 Data Format in Memory

zero-extended in logical operations (TST, AND, OR, and XOR instructions).

Load/Store Architecture: The SH7709S features a load-store architecture in which ba operations are executed in registers. Operations requiring memory access are executed following register loading, except for bit-manipulation operations such as logical AND which are executed directly in memory.

Delayed Branching: Unconditional branching is implemented as delayed branch opera Pipeline disruptions due to branching are minimized by the execution of the instruction the delayed branch instruction prior to branching. Conditional branch instructions are o kinds, delayed and normal.

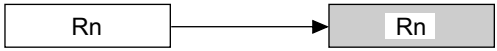
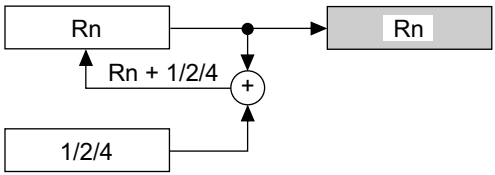
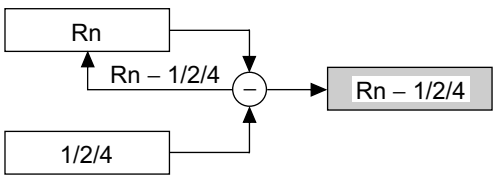
```
BRA     TRGET
ADD     R1, R0      ;ADD is executed prior to branching to TRGET
```

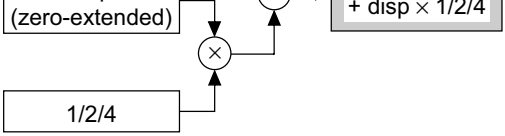
Literals: Byte-length literals are inserted directly into the instruction code as immedia
maintain the 16-bit fixed-length instruction code, word or longword literals are stored
main memory rather than inserted directly into the instruction code. The memory table
by the MOV instruction using PC-relative addressing with displacement, as follows:

```
MOV.W    @(disp, PC), R0
```

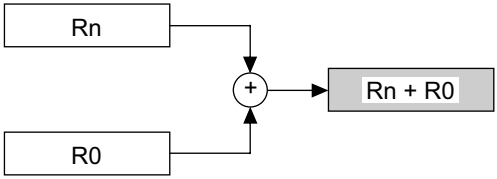
Absolute Addresses: As with word and longword literals, absolute addresses must als
in a table in main memory. The value of the absolute address is transferred to a registe
operand access is specified by indexed register-indirect addressing, with the absolute a
loaded (like word and longword immediate data) during instruction execution.

16-Bit and 32-Bit Displacements: In the same way, 16-bit and 32-bit displacements a
stored in a table in main memory. Exactly like absolute addresses, the displacement va
transferred to a register and the operand access is specified by indexed register-indirec
loading the displacement (like word and longword immediate data) during instruction

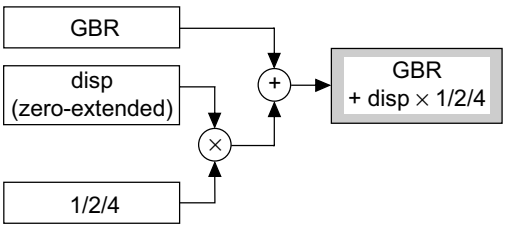
Register indirect	@Rn	Effective address is register Rn contents. 	Rn
Register indirect with post-increment	@Rn+	Effective address is register Rn contents. A constant is added to Rn after instruction execution: 1 for a byte operand, 2 for a word operand, 4 for a longword operand. 	Rn After instruc execution Byte: Rn + 1 Word: Rn + Longword: F
Register indirect with pre-decrement	@–Rn	Effective address is register Rn contents, decremented by a constant beforehand: 1 for a byte operand, 2 for a word operand, 4 for a longword operand. 	Byte: Rn – 1 Word: Rn – Longword: F (Instruction with Rn afte calculation)



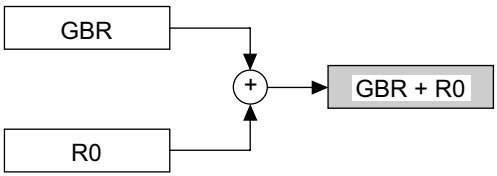
Indexed register indirect	@(R0, Rn)	Effective address is sum of register Rn and R0 contents.	Rn + R0

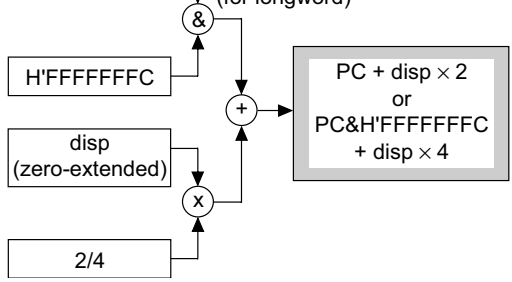


GBR indirect with displacement	@(disp:8, GBR)	Effective address is register GBR contents with 8-bit displacement disp added. After disp is zero-extended, it is multiplied by 1 (byte), 2 (word), or 4 (longword), according to the operand size.	Byte: GBR Word: GBR Longword: × 4

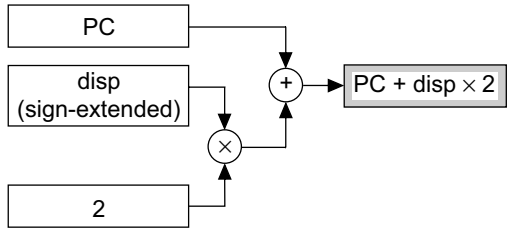


Indexed GBR indirect	@(R0, GBR)	Effective address is sum of register GBR and R0 contents.	GBR + R0

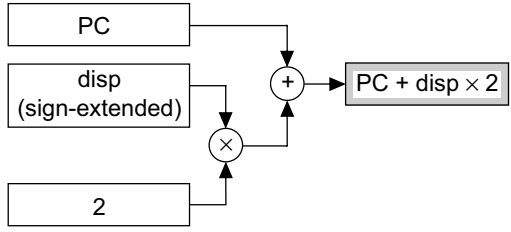




PC-relative	disp:8	Effective address is register PC contents with 8-bit displacement disp added after being sign-extended and multiplied by 2.	PC + disp ×



	disp:12	Effective address is register PC contents with 12-bit displacement disp added after being sign-extended and multiplied by 2.	PC + disp ×



Immediate	#imm:8	8-bit immediate data imm of TST, AND, OR, or XOR instruction is zero-extended.	—
	#imm:8	8-bit immediate data imm of MOV, ADD, or CMP/EQ instruction is sign-extended.	—
	#imm:8	8-bit immediate data imm of TRAPA instruction is zero-extended and multiplied by 4.	—

Note: For the addressing modes below that use a displacement (disp), the assembler in this manual show the value before scaling (×1, ×2, or ×4) is performed accor operand size. This is done to clarify the operation of the IC. Refer to the relevan notation rules for the actual assembler descriptions.

@ (disp:4, Rn) ; Register indirect with displacement

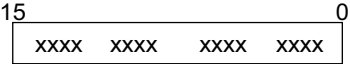
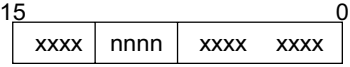
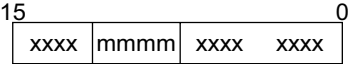
@ (disp:8, Rn) ; GBR indirect with displacement

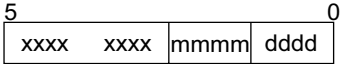
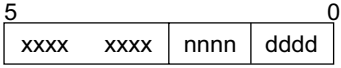
@ (disp:8, PC) ; PC-relative with displacement

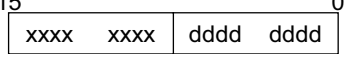
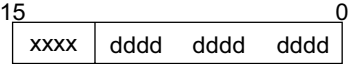
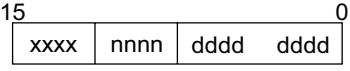
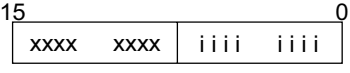
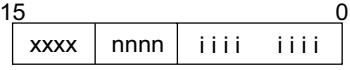
disp:8, disp:12; PC-relative

dddd: Displacement

Table 2.3 Instruction Formats

Instruction Format	Source Operand	Destination Operand	In Ex
0 format 	—	—	N
n format 	—	nnnn: register direct	M
	Control register or system register	nnnn: register direct	S M
	Control register or system register	nnnn: register indirect with pre-decrement	S S
m format 	mmmm: register direct	Control register or system register	LD Rn
	mmmm: register indirect with post-increment	Control register or system register	LD @
	mmmm: register indirect	—	JM
	mmmm: PC-relative using Rm	—	BR

	(multiply-and-accumulate operation) nnnn: * register indirect with post-increment (multiply-and-accumulate operation)		
	mmmm: register indirect with post-increment	nnnn: register direct	M @
	mmmm: register direct	nnnn: register indirect with pre-decrement	M R
	mmmm: register direct	nnnn: indexed register indirect	M R
md format 	mmmmdddd: register indirect with displacement	R0 (register direct)	M @
nd4 format 	R0 (register direct)	nnnndddd: register indirect with displacement	M R

d format		dddddddd: GBR indirect with displacement	R0 (register direct)	MO @(d 0
		R0 (register direct)	ddddddddd: GBR indirect with displacement	MO R0,)
		ddddddddd: PC-relative with displacement	R0 (register direct)	MO @(d
		ddddddddd: PC-relative	—	BF
d12 format		ddddddddddddd: PC-relative	—	BRA (lab PC)
nd8 format		ddddddddd: PC-relative with displacement	nnnn: register direct	MO @(d
i format		iiiiiiii: immediate	Indexed GBR indirect	AND #im @(R
		iiiiiiii: immediate	R0 (register direct)	AND #im
		iiiiiiii: immediate	—	TRA
ni format		iiiiiiii: immediate	nnnn: register direct	ADD #im

Note: * In a multiply-and-accumulate instruction, nnnn is the source register.

Classification	Types	Code	Function	I
Data transfer	5	MOV	Data transfer	3
		MOVA	Effective address transfer	
		MOVT	T bit transfer	
		SWAP	Swap of upper and lower bytes	
		XTRCT	Extraction of middle of linked registers	
Arithmetic operations	21	ADD	Binary addition	3
		ADDC	Binary addition with carry	
		ADDV	Binary addition with overflow check	
		CMP/cond	Comparison	
		DIV1	Division	
		DIV0S	Initialization of signed division	
		DIV0U	Initialization of unsigned division	
		DMULS	Signed double-precision multiplication	
		DMULU	Unsigned double-precision multiplication	
		DT	Decrement and test	
		EXTS	Sign extension	
		EXTU	Zero extension	
		MAC	Multiply-and-accumulate operation, double-precision multiply-and-accumulate operation	

		SUB	Binary subtraction	
		SUBC	Binary subtraction with borrow	
		SUBV	Binary subtraction with underflow check	
Logic operations	6	AND	Logical AND	14
		NOT	Bit inversion	
		OR	Logical OR	
		TAS	Memory test and bit set	
		TST	Logical AND and T bit set	
		XOR	Exclusive OR	
Shift	12	ROTL	One-bit left rotation	16
		ROTR	One-bit right rotation	
		ROTCL	One-bit left rotation with T bit	
		ROTCR	One-bit right rotation with T bit	
		SHAL	One-bit arithmetic left shift	
		SHAR	One-bit arithmetic right shift	
		SHLL	One-bit logical left shift	
		SHLLn	n-bit logical left shift	
		SHLR	One-bit logical right shift	
		SHLRn	n-bit logical right shift	
		SHAD	Dynamic arithmetic shift	
		SHLD	Dynamic logical shift	

		BSRF	Branch to subroutine procedure	
		JMP	Unconditional branch	
		JSR	Branch to subroutine procedure	
		RTS	Return from subroutine procedure	
System control	15	CLRMAC	MAC register clear	7
		CLRT	Clear T bit	
		CLRS	Clear S bit	
		LDC	Load to control register	
		LDS	Load to system register	
		LDTLB	Load PTE to TLB	
		NOP	No operation	
		PREF	Prefetch data to cache	
		RTE	Return from exception handling	
		SETS	Set S bit	
		SETT	Set T bit	
		SLEEP	Shift to power-down mode	
		STC	Store from control register	
		STS	Store from system register	
		TRAPA	Trap exception handling	
Total: 68				1

		Rn: Destination register imm: Immediate data disp: Displacement
Instruction code	MSB ↔ LSB	mmmm: Source register nnnn: Destination register 0000: R0 0001: R1 ..... 1111: R15 iiii: Immediate data dddd: Displacement*
Operation summary	→, ← (xx) M/Q/T & | ^ ~ <<n, >>n	Direction of transfer Memory operand Flag bits in SR Logical AND of each bit Logical OR of each bit Exclusive OR of each bit Logical NOT of each bit n-bit shift
Privileged mode		Indicates whether privileged mode applies
Execution cycles		Value when no wait states are inserted The execution cycles listed in the table are minim actual number of cycles may be increased in cas the following: 1. When contention occurs between instruction f data access 2. When the destination register of the load instr (memory → register) and the register used by instruction are the same
T bit		Value of T bit after instruction is executed —: No change

Note: * Scaling (×1, ×2, ×4) is performed according to the instruction operand size.

MOV.L	@(disp,PC),Rn	(disp × 4 + PC) → Rn	1101nnnndddddddd	—
MOV	Rm,Rn	Rm → Rn	0110nnnnmmmm0011	—
MOV.B	Rm,@Rn	Rm → (Rn)	0010nnnnmmmm0000	—
MOV.W	Rm,@Rn	Rm → (Rn)	0010nnnnmmmm0001	—
MOV.L	Rm,@Rn	Rm → (Rn)	0010nnnnmmmm0010	—
MOV.B	@Rm,Rn	(Rm) → Sign extension → Rn	0110nnnnmmmm0000	—
MOV.W	@Rm,Rn	(Rm) → Sign extension → Rn	0110nnnnmmmm0001	—
MOV.L	@Rm,Rn	(Rm) → Rn	0110nnnnmmmm0010	—
MOV.B	Rm,@–Rn	Rn–1 → Rn, Rm → (Rn)	0010nnnnmmmm0100	—
MOV.W	Rm,@–Rn	Rn–2 → Rn, Rm → (Rn)	0010nnnnmmmm0101	—
MOV.L	Rm,@–Rn	Rn–4 → Rn, Rm → (Rn)	0010nnnnmmmm0110	—
MOV.B	@Rm+,Rn	(Rm) → Sign extension → Rn, Rm + 1 → Rm	0110nnnnmmmm0100	—
MOV.W	@Rm+,Rn	(Rm) → Sign extension → Rn, Rm + 2 → Rm	0110nnnnmmmm0101	—
MOV.L	@Rm+,Rn	(Rm) → Rn,Rm + 4 → Rm	0110nnnnmmmm0110	—
MOV.B	R0,@(disp,Rn)	R0 → (disp + Rn)	10000000nnnndddd	—
MOV.W	R0,@(disp,Rn)	R0 → (disp × 2 + Rn)	10000001nnnndddd	—
MOV.L	Rm,@(disp,Rn)	Rm → (disp × 4 + Rn)	0001nnnnmmmmdddd	—
MOV.B	@(disp,Rm),R0	(disp + Rm) → Sign extension → R0	10000100mmmmdddd	—
MOV.W	@(disp,Rm),R0	(disp × 2 + Rm) → Sign extension → R0	10000101mmmmdddd	—
MOV.L	@(disp,Rm),Rn	(disp × 4 + Rm) → Rn	0101nnnnmmmmdddd	—
MOV.B	Rm,@(R0,Rn)	Rm → (R0 + Rn)	0000nnnnmmmm0100	—

MOV.B	R0,@(disp,GBR)	R0 → (disp + GBR)	11000000dddddddd	—	1
MOV.W	R0,@(disp,GBR)	R0 → (disp × 2 + GBR)	11000001dddddddd	—	1
MOV.L	R0,@(disp,GBR)	R0 → (disp × 4 + GBR)	11000010dddddddd	—	1
MOV.B	@(disp,GBR),R0	(disp + GBR) → Sign extension → R0	11000100dddddddd	—	1
MOV.W	@(disp,GBR),R0	(disp × 2 + GBR) → Sign extension → R0	11000101dddddddd	—	1
MOV.L	@(disp,GBR),R0	(disp × 4 + GBR) → R0	11000110dddddddd	—	1
MOVA	@(disp,PC),R0	disp × 4 + PC → R0	11000111dddddddd	—	1
MOVT	Rn	T → Rn	0000nnnn00101001	—	1
SWAP.B	Rm,Rn	Rm → Swap the bottom two bytes → Rn	0110nnnnmmmm1000	—	1
SWAP.W	Rm,Rn	Rm → Swap two consecutive words → Rn	0110nnnnmmmm1001	—	1
XTRCT	Rm,Rn	Rm: Middle 32 bits of Rn → Rn	0010nnnnmmmm1101	—	1

		Carry → T			
ADDV	Rm,Rn	Rn + Rm → Rn, Overflow → T	0011nnnnmmmm1111	—	1
CMP/EQ	#imm,R0	If R0 = imm, 1 → T	10001000iiiiiiii	—	1
CMP/EQ	Rm,Rn	If Rn = Rm, 1 → T	0011nnnnmmmm0000	—	1
CMP/HS	Rm,Rn	If Rn ≥ Rm with unsigned data, 1 → T	0011nnnnmmmm0010	—	1
CMP/GE	Rm,Rn	If Rn ≥ Rm with signed data, 1 → T	0011nnnnmmmm0011	—	1
CMP/HI	Rm,Rn	If Rn > Rm with unsigned data, 1 → T	0011nnnnmmmm0110	—	1
CMP/GT	Rm,Rn	If Rn > Rm with signed data, 1 → T	0011nnnnmmmm0111	—	1
CMP/PZ	Rn	If Rn ≥ 0, 1 → T	0100nnnn00010001	—	1
CMP/PL	Rn	If Rn > 0, 1 → T	0100nnnn00010101	—	1
CMP/STR	Rm,Rn	If Rn and Rm have an equivalent byte, 1 → T	0010nnnnmmmm1100	—	1
DIV1	Rm,Rn	Single-step division (Rn/Rm)	0011nnnnmmmm0100	—	1
DIV0S	Rm,Rn	MSB of Rn → Q, MSB of Rm → M, M ^ Q → T	0010nnnnmmmm0111	—	1
DIV0U		0 → M/Q/T	0000000000011001	—	1

		0, 1 → T, else 0 → T			
EXTS.B	Rm,Rn	A byte in Rm is sign-extended → Rn	0110nnnnmmmm1110	—	1
EXTS.W	Rm,Rn	A word in Rm is sign-extended → Rn	0110nnnnmmmm1111	—	1
EXTU.B	Rm,Rn	A byte in Rm is zero-extended → Rn	0110nnnnmmmm1100	—	1
EXTU.W	Rm,Rn	A word in Rm is zero-extended → Rn	0110nnnnmmmm1101	—	1
MAC.L	@Rm+,@Rn+	Signed operation of (Rn) × (Rm) + MAC → MAC, Rn + 4 → Rn, Rm + 4 → Rm, 32 × 32 + 64 → 64 bits	0000nnnnmmmm1111	—	2(to 5)*
MAC.W	@Rm+,@Rn+	Signed operation of (Rn) × (Rm) + MAC → MAC, Rn + 2 → Rn, Rm + 2 → Rm, 16 × 16 + 64 → 64 bits	0100nnnnmmmm1111	—	2(to 5)*
MUL.L	Rm,Rn	Rn × Rm → MACL, 32 × 32 → 32 bits	0000nnnnmmmm0111	—	2(to 5)*
MULS.W	Rm,Rn	Signed operation of Rn × Rm → MACL, 16 × 16 → 32 bits	0010nnnnmmmm1111	—	1(to 3)*
MULU.W	Rm,Rn	Unsigned operation of Rn × Rm → MACL, 16 × 16 → 32 bits	0010nnnnmmmm1110	—	1(to 3)*

SUB	Rm,Rn	Rn−Rm → Rn, Underflow → T	0011nnnnmmmm1000	1

Note: * The normal number of execution cycles is shown. The value in parentheses is
of cycles required in case of contention with the preceding or following instruc

		(R0 + GBR)			
NOT	Rm,Rn	~Rm → Rn	0110nnnnmmmm0111	—	1
OR	Rm,Rn	Rn | Rm → Rn	0010nnnnmmmm1011	—	1
OR	#imm,R0	R0 | imm → R0	11001011iiiiiiii	—	1
OR.B	#imm,@(R0,GBR)	(R0 + GBR) | imm → (R0 + GBR)	11001111iiiiiiii	—	3
TAS.B	@Rn	If (Rn) is 0, 1 → T; 1 → MSB of (Rn)	0100nnnn00011011	—	3
TST	Rm,Rn	Rn & Rm; if the result is 0, 1 → T	0010nnnnmmmm1000	—	1
TST	#imm,R0	R0 & imm; if the result is 0, 1 → T	11001000iiiiiiii	—	1
TST.B	#imm,@(R0,GBR)	(R0 + GBR) & imm; if the result is 0, 1 → T	11001100iiiiiiii	—	3
XOR	Rm,Rn	Rn ^ Rm → Rn	0010nnnnmmmm1010	—	1
XOR	#imm,R0	R0 ^ imm → R0	11001010iiiiiiii	—	1
XOR.B	#imm,@(R0,GBR)	(R0 + GBR) ^ imm → (R0 + GBR)	11001110iiiiiiii	—	3

ROTCR	Rn	T → Rn → T	0100nnnn00100101	—	1
SHAD	Rm,Rn	Rn ≥ 0: Rn << Rm → Rn Rn < 0: Rn >> Rm → [MSB → Rn]	0100nnnnmmmm1100	—	1
SHAL	Rn	T ← Rn ← 0	0100nnnn00100000	—	1
SHAR	Rn	MSB → Rn → T	0100nnnn00100001	—	1
SHLD	Rm,Rn	Rn ≥ 0: Rn << Rm → Rn Rn < 0: Rn >> Rm → [0 → Rn]	0100nnnnmmmm1101	—	1
SHLL	Rn	T ← Rn ← 0	0100nnnn00000000	—	1
SHLR	Rn	0 → Rn → T	0100nnnn00000001	—	1
SHLL2	Rn	Rn << 2 → Rn	0100nnnn00001000	—	1
SHLR2	Rn	Rn >> 2 → Rn	0100nnnn00001001	—	1
SHLL8	Rn	Rn << 8 → Rn	0100nnnn00011000	—	1
SHLR8	Rn	Rn >> 8 → Rn	0100nnnn00011001	—	1
SHLL16	Rn	Rn << 16 → Rn	0100nnnn00101000	—	1
SHLR16	Rn	Rn >> 16 → Rn	0100nnnn00101001	—	1

		if T = 1, nop			
BT	label	if T = 1, disp × 2 + PC → PC; if T = 0, nop	10001001dddddddd	—	3/1
BT/S	label	Delayed branch, If T = 1, disp × 2 + PC → PC; if T = 0, nop	10001101dddddddd	—	2/1
BRA	label	Delayed branch, disp × 2 + PC → PC	1010dddddddddddd	—	2
BRAF	Rm	Delayed branch, Rm + PC → PC	0000mmmm00100011	—	2
BSR	label	Delayed branch, PC → PR, disp × 2 + PC → PC	1011dddddddddddd	—	2
BSRF	Rm	Delayed branch, PC → PR, Rm + PC → PC	0000mmmm00000011	—	2
JMP	@Rm	Delayed branch, Rm → PC	0100mmmm00101011	—	2
JSR	@Rm	Delayed branch, PC → PR, Rm → PC	0100mmmm00001011	—	2
RTS		Delayed branch, PR → PC	0000000000001011	—	2

Note: * One state when there is no branch.

LDC	Rm,SR	Rm → SR	0100mmmm00001110	√
LDC	Rm,GBR	Rm → GBR	0100mmmm00011110	—
LDC	Rm,VBR	Rm → VBR	0100mmmm00101110	√
LDC	Rm,SSR	Rm → SSR	0100mmmm00111110	√
LDC	Rm,SPC	Rm → SPC	0100mmmm01001110	√
LDC	Rm,R0_BANK	Rm → R0_BANK	0100mmmm10001110	√
LDC	Rm,R1_BANK	Rm → R1_BANK	0100mmmm10011110	√
LDC	Rm,R2_BANK	Rm → R2_BANK	0100mmmm10101110	√
LDC	Rm,R3_BANK	Rm → R3_BANK	0100mmmm10111110	√
LDC	Rm,R4_BANK	Rm → R4_BANK	0100mmmm11001110	√
LDC	Rm,R5_BANK	Rm → R5_BANK	0100mmmm11011110	√
LDC	Rm,R6_BANK	Rm → R6_BANK	0100mmmm11101110	√
LDC	Rm,R7_BANK	Rm → R7_BANK	0100mmmm11111110	√
LDC.L	@Rm+,SR	(Rm) → SR, Rm + 4 → Rm	0100mmmm00000111	√
LDC.L	@Rm+,GBR	(Rm) → GBR, Rm + 4 → Rm	0100mmmm00010111	—
LDC.L	@Rm+,VBR	(Rm) → VBR, Rm + 4 → Rm	0100mmmm00100111	√
LDC.L	@Rm+,SSR	(Rm) → SSR, Rm + 4 → Rm	0100mmmm00110111	√
LDC.L	@Rm+,SPC	(Rm) → SPC, Rm + 4 → Rm	0100mmmm01000111	√
LDC.L	@Rm+, R0_BANK	(Rm) → R0_BANK, Rm + 4 → Rm	0100mmmm10000111	√
LDC.L	@Rm+, R1_BANK	(Rm) → R1_BANK, Rm + 4 → Rm	0100mmmm10010111	√
LDC.L	@Rm+, R2_BANK	(Rm) → R2_BANK, Rm + 4 → Rm	0100mmmm10100111	√
LDC.L	@Rm+, R3_BANK	(Rm) → R3_BANK, Rm + 4 → Rm	0100mmmm10110111	√
LDC.L	@Rm+, R4_BANK	(Rm) → R4_BANK, Rm + 4 → Rm	0100mmmm11000111	√
LDC.L	@Rm+, R5_BANK	(Rm) → R5_BANK, Rm + 4 → Rm	0100mmmm11010111	√

LDS.L	@Rm+,MACH	(Rm) → MACH, Rm + 4 → Rm	0100mmmm00000110	—	1
LDS.L	@Rm+,MACL	(Rm) → MACL, Rm + 4 → Rm	0100mmmm00010110	—	1
LDS.L	@Rm+,PR	(Rm) → PR, Rm + 4 → Rm	0100mmmm00100110	—	1
LDTLB		PTEH/PTEL → TLB	0000000000111000	√	1
NOP		No operation	0000000000001001	—	1
PREF	@Rm	(Rm) → cache	0000mmmm10000011	—	2
RTE		Delayed branch, SSR → SR, SPC → PC	0000000000101011	√	4
SETS		1 → S	0000000001011000	—	1
SETT		1 → T	0000000000011000	—	1
SLEEP		Sleep	0000000000011011	√	4
STC	SR,Rn	SR → Rn	0000nnnn00000010	√	1
STC	GBR,Rn	GBR → Rn	0000nnnn00010010	—	1
STC	VBR,Rn	VBR → Rn	0000nnnn00100010	√	1
STC	SSR,Rn	SSR → Rn	0000nnnn00110010	√	1
STC	SPC,Rn	SPC → Rn	0000nnnn01000010	√	1
STC	R0_BANK,Rn	R0_BANK→ Rn	0000nnnn10000010	√	1
STC	R1_BANK,Rn	R1_BANK→ Rn	0000nnnn10010010	√	1
STC	R2_BANK,Rn	R2_BANK→ Rn	0000nnnn10100010	√	1
STC	R3_BANK,Rn	R3_BANK→ Rn	0000nnnn10110010	√	1
STC	R4_BANK,Rn	R4_BANK→ Rn	0000nnnn11000010	√	1
STC	R5_BANK,Rn	R5_BANK→ Rn	0000nnnn11010010	√	1
STC	R6_BANK,Rn	R6_BANK→ Rn	0000nnnn11100010	√	1
STC	R7_BANK,Rn	R7_BANK→ Rn	0000nnnn11110010	√	1
STC.L	SR,@–Rn	Rn–4 → Rn, SR → (Rn)	0100nnnn00000011	√	2
STC.L	GBR,@–Rn	Rn–4 → Rn, GBR → (Rn)	0100nnnn00010011	—	2
STC.L	VBR,@–Rn	Rn–4 → Rn, VBR → (Rn)	0100nnnn00100011	√	2

Note: * The number of cycles until the sleep state is entered.

	@–Rn			
STC.L	R3_BANK, @–Rn	Rn–4 → Rn, R3_BANK → (Rn)	0100nnnn10110011	√
STC.L	R4_BANK, @–Rn	Rn–4 → Rn, R4_BANK → (Rn)	0100nnnn11000011	√
STC.L	R5_BANK, @–Rn	Rn–4 → Rn, R5_BANK → (Rn)	0100nnnn11010011	√
STC.L	R6_BANK, @–Rn	Rn–4 → Rn, R6_BANK → (Rn)	0100nnnn11100011	√
STC.L	R7_BANK, @–Rn	Rn–4 → Rn, R7_BANK → (Rn)	0100nnnn11110011	√
STS	MACH,Rn	MACH → Rn	0000nnnn00001010	—
STS	MACL,Rn	MACL → Rn	0000nnnn00011010	—
STS	PR,Rn	PR → Rn	0000nnnn00101010	—
STS.L	MACH,@–Rn	Rn–4 → Rn, MACH → (Rn)	0100nnnn00000010	—
STS.L	MACL,@–Rn	Rn–4 → Rn, MACL → (Rn)	0100nnnn00010010	—
STS.L	PR,@–Rn	Rn–4 → Rn, PR → (Rn)	0100nnnn00100010	—
TRAPA	#imm	PC → SPC, SR → SSR, imm → TRA	11000011iiiiiiii	—

Notes: 1. The table shows the minimum number of execution cycles. The actual num
instruction execution cycles will increase in cases such as the following:
- When there is contention between an instruction fetch and data access
- When the destination register in a load (memory-to-register) instruction i
by the next instruction

2. With the addressing modes using displacement (disp) listed below, the ass
descriptions in this manual show the value before scaling (×1, ×2, or ×4) is
This is done to clarify the operation of the chip. For the actual assembler de
refer to the individual assembler notation rules.

@ (disp:4, Rn) ; Register-indirect with displacement
@ (disp:8, Rn) ; GBR-indirect with displacement
@ (disp:8, PC) ; PC-relative with displacement
disp:8, disp:12 ; PC-relative

0000	Rn	Fx	0001									
0000	Rn	00MD	0010	STC	SR,Rn	STC GBR,Rn			STC VBR,Rn			STC SS
0000	Rn	01MD	0010	STC	SPC,Rn							
0000	Rn	10MD	0010	STC	R0_BANK,Rn	STC	R1_BANK,Rn		STC	R2_BANK,Rn		STC
0000	Rn	11MD	0010	STC	R4_BANK,Rn	STC	R5_BANK,Rn		STC	R6_BANK,Rn		STC
0000	Rm	00MD	0011	BSRF	Rm				BRAF	Rm		
0000	Rn	10MD	0011	PREF	@Rn							
0000	Rn	Rm	01MD	MOV.B	Rm,@(R0,Rn)	MOV.W	Rm,@(R0,Rn)		MOV.L	Rm,@(R0,Rn)		MUL.L
0000	0000	00MD	1000	CLRT		SETT			CLRMAC			LDTLB
0000	0000	01MD	1000	CLRS		SETS						
0000	0000	Fx	1001	NOP		DIV0U						
0000	0000	Fx	1010									
0000	0000	Fx	1011	RTS		SLEEP			RTE			
0000	Rn	Fx	1000									
0000	Rn	Fx	1001						MOVT	Rn		
0000	Rn	Fx	1010	STS	MACH,Rn	STS	MACL,Rn		STS	PR,Rn		
0000	Rn	Fx	1011									
0000	Rn	Rm	11MD	MOV.B	@(R0,Rm),Rn	MOV.W	@(R0,Rm),Rn		MOV.L	@(R0,Rm),Rn		MAC.L
0001	Rn	Rm	disp	MOV.L	Rm,@(disp:4,Rn)							
0010	Rn	Rm	00MD	MOV.B	Rm,@Rn	MOV.W	Rm,@Rn		MOV.L	Rm,@Rn		
0010	Rn	Rm	01MD	MOV.B	Rm,@-Rn	MOV.W	Rm,@-Rn		MOV.L	Rm,@-Rn		DIV0S
0010	Rn	Rm	10MD	TST	Rm,Rn	AND	Rm,Rn		XOR	Rm,Rn		OR
0010	Rn	Rm	11MD	CMP/STR	Rm,Rn	XTRCT	Rm,Rn		MULU.W	Rm,Rn		MULSW
0011	Rn	Rm	00MD	CMP/EQ	Rm,Rn				CMP/HS	Rm,Rn		CMP/G
0011	Rn	Rm	01MD	DIV1	Rm,Rn	DMULU.L	Rm,Rn		CMP/HI	Rm,Rn		CMP/G
0011	Rn	Rm	10MD	SUB	Rm,Rn				SUBC	Rm,Rn		SUBV
0011	Rn	Rm	11MD	ADD	Rm,Rn	DMULS.L	Rm,Rn		ADDC	Rm,Rn		ADDV

0100	Rn	11MD	0011	STC.L	R4_BANK,@-Rn	STC.L	R5_BANK,@-Rn	STC.L	R6_BANK,@-Rn	STC.L
0100	Rn	Fx	0100	ROTL	Rn			ROTCL	Rn	
0100	Rn	Fx	0101	ROTR	Rn	CMP/PL	Rn	ROTCR	Rn	
0100	Rm	Fx	0110	LDS.L	@Rm+,MACH	LDS.L	@Rm+,MACL	LDS.L	@Rm+,PR	
0100	Rm	00MD	0111	LDC.L	@Rm+,SR	LDC.L	@Rm+,GBR	LDC.L	@Rm+,VBR	LDC.L
0100	Rm	01MD	0111	LDC.L	@Rm+,SPC					
0100	Rm	10MD	0111	LDC.L	@Rm+,R0_BANK	LDC.L	@Rm+,R1_BANK	LDC.L	@Rm+,R2_BANK	LDC.L
0100	Rm	11MD	0111	LDC.L	@Rm+,R4_BANK	LDC.L	@Rm+,R5_BANK	LDC.L	@Rm+,R6_BANK	LDC.L
0100	Rn	Fx	1000	SHLL2	Rn	SHLL8	Rn	SHLL16	Rn	
0100	Rn	Fx	1001	SHLR2	Rn	SHLR8	Rn	SHLR16	Rn	
0100	Rm	Fx	1010	LDS	Rm,MACH	LDS	Rm,MACL	LDS	Rm,PR	
0100	Rm/ Rn	Fx	1011	JSR	@Rm	TAS.B	@Rn	JMP	@Rm	
0100	Rn	Rm	1100	SHAD	Rm,Rn					
0100	Rn	Rm	1101	SHLD	Rm,Rn					
0100	Rm	00MD	1110	LDC	Rm,SR	LDC	Rm,GBR	LDC	Rm,VBR	LDC
0100	Rm	01MD	1110	LDC	Rm,SPC					
0100	Rm	10MD	1110	LDC	Rm,R0_BANK	LDC	Rm,R1_BANK	LDC	Rm,R2_BANK	LDC
0100	Rm	11MD	1110	LDC	Rm,R4_BANK	LDC	Rm,R5_BANK	LDC	Rm,R6_BANK	LDC
0100	Rn	Rm	1111	MAC.W	@Rm+,@Rn+					
0101	Rn	Rm	disp	MOV.L	@(disp:4,Rm),Rn					
0110	Rn	Rm	00MD	MOV.B	@Rm,Rn	MOV.W	@Rm,Rn	MOV.L	@Rm,Rn	MOV
0110	Rn	Rm	01MD	MOV.B	@Rm+,Rn	MOV.W	@Rm+,Rn	MOV.L	@Rm+,Rn	NOT
0110	Rn	Rm	10MD	SWAP.B	Rm,Rn	SWAP.W	Rm,Rn	NEGC	Rm,Rn	NEG
0110	Rn	Rm	11MD	EXTU.B	Rm,Rn	EXTU.W	Rm,Rn	EXTS.B	Rm,Rn	EXTS
0111	Rn	imm		ADD	#imm:8,Rn					

1010	disp		BRA label:12			
1011	disp		BSR label:12			
1100	00MD	imm/disp	MOV.B R0,@(disp:8,GBR)	MOV.W R0,@(disp:8,GBR)	MOV.L R0,@(disp:8,GBR)	TRAPA
1100	01MD	disp	MOV.B @(disp:8,GBR),R0	MOV.W @(disp:8,GBR),R0	MOV.L @(disp:8,GBR),R0	MOVA @(dis
1100	10MD	imm	TST #imm:8,R0	AND #imm:8,R0	XOR #imm:8,R0	OR
1100	11MD	imm	TST.B #imm:8,@(R0,GBR)	AND.B #imm:8,@(R0,GBR)	XOR.B #imm:8,@(R0,GBR)	OR.B #imm:
1101	Rn	disp	MOV.L @(disp:8,PC),Rn			
1110	Rn	imm	MOV #imm:8,Rn			
1111	**********					

Note: See the SH-3/SH-3E/SH3-DSP Programming Manual for details.

for more information on resets.

In the power-on reset state, the internal states of the CPU and the on-chip supporting r
registers are initialized. In the manual reset state, the internal states of the CPU and re
chip supporting modules other than the bus state controller (BSC) are initialized. Refe
register configurations in the relevant sections for further details.

Exception-Handling State: This is a transient state during which the CPU's processo
is altered by a reset, general exception, or interrupt exception handling.

In the case of a reset, the CPU branches to address H'A0000000 and starts executing t
coded exception handling program.

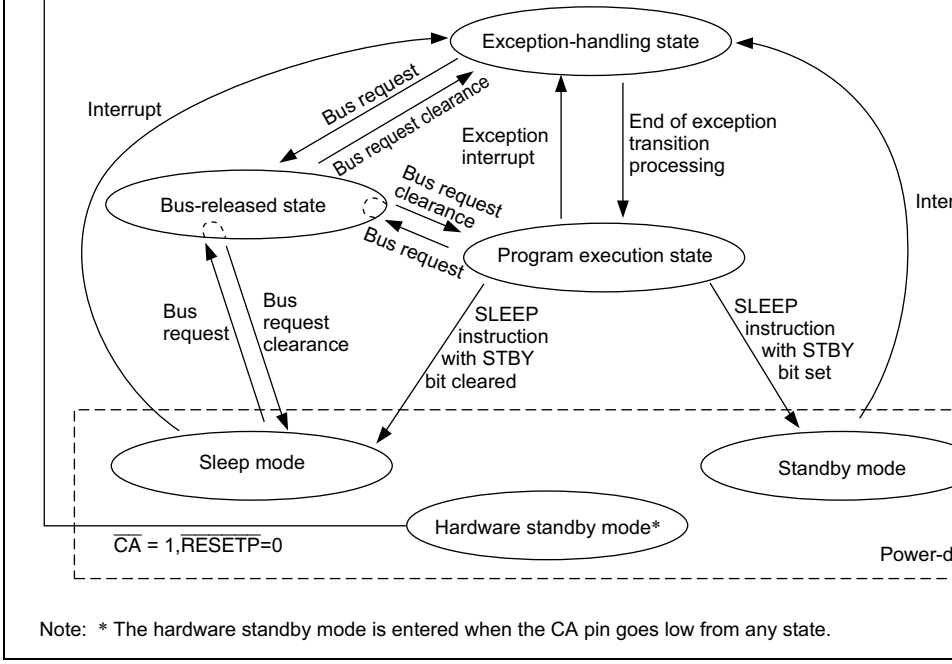
In the case of a general exception or interrupt, the program counter (PC) contents are s
saved program counter (SPC) and the status register (SR) contents are saved in the sav
register (SSR). The CPU branches to the start address of the user-coded exception ser
found from the sum of the contents of the vector base address and the vector offset. Se
Exception Processing, for more information on resets, general exceptions, and interrup

Program Execution State: In this state the CPU executes program instructions in seq

Power-Down State: In the power-down state, CPU operation halts and power consum
reduced. There are two modes in the power-down state: sleep mode, and standby mod
section 8, Power-Down Modes, for more information.

Bus-Released State: In this state the CPU has released the bus to a device that reques

Transitions between the states are shown in figure 2.8.



Note: * The hardware standby mode is entered when the CA pin goes low from any state.

Figure 2.8 Processor State Transitions

2.5.2 Processor Modes

There are two processor modes: privileged mode and user mode. The processor mode is determined by the processor mode bit (MD) in the status register (SR). User mode is sel when the MD bit is 0, and privileged mode when the MD bit is 1. When the reset state o exception state is entered, the MD bit is set to 1. When exception handling ends, the MD cleared to 0 and user mode is entered. There are certain registers and bits which can onl accessed in privileged mode.

translation. The SH7055 features a resident translation look-aside buffer (TLB) that
information for user-created address translation tables located in external memory. It e
speed translation of virtual addresses into physical addresses. Address translation uses
system and supports two page sizes (1 kbytes and 4 kbytes). The access right to virtua
space can be set for privileged and user modes to provide memory protection.

3.1.2 Role of MMU

The MMU is a feature designed to make efficient use of physical memory. As shown
if a process is smaller in size than the physical memory, the entire process can be map
physical memory. However, if the process increases in size to the extent that it no long
physical memory, it becomes necessary to partition the process and to map those parts
execution onto memory as occasion demands ((1)). Having the process itself consider
mapping onto physical memory would impose a large burden on the process. To lighte
burden, the idea of virtual memory was born as a means of performing en bloc mappin
physical memory ((2)). In a virtual memory system, substantially more virtual memor
physical memory is provided, and the process is mapped onto this virtual memory. Th
only has to consider operation in virtual memory. Mapping from virtual memory to ph
memory is handled by the MMU. The MMU is normally controlled by the operating s
switching physical memory to allow the virtual memory required by a process to be m
physical memory in a smooth fashion. Switching of physical memory is carried out via
storage, etc.

The virtual memory system that came into being in this way is particularly effective in
sharing system (TSS) in which a number of processes are running simultaneously ((3))
processes running in a TSS had to take mapping onto virtual memory into considerati
running, it would not be possible to increase efficiency. Virtual memory is thus used t
load on the individual processes and so improve efficiency ((4)). In the virtual memor
virtual memory is allocated to each process. The task of the MMU is to perform effici
of these virtual memory areas onto physical memory. It also has a memory protection
prevents one process from inadvertently accessing another process's physical memory

When address translation from virtual memory to physical memory is performed using
it may occur that the relevant translation information is not recorded in the MMU, wit
that one process may inadvertently access the virtual memory allocated to another pro

address translation information is normally performed by software. This makes it possi
memory management to be performed flexibly by software.

The MMU has two methods of mapping from virtual memory to physical memory: a pa
method using fixed-length address translation, and a segment method using variable-le
address translation. With the paging method, the unit of translation is a fixed-size addre
(usually of 1 to 64 kbytes) called a page. This LSI uses the paging method.

In the following text, the SH7709S address space in virtual memory is referred to as vi
address space, and address space in physical memory as physical memory space.

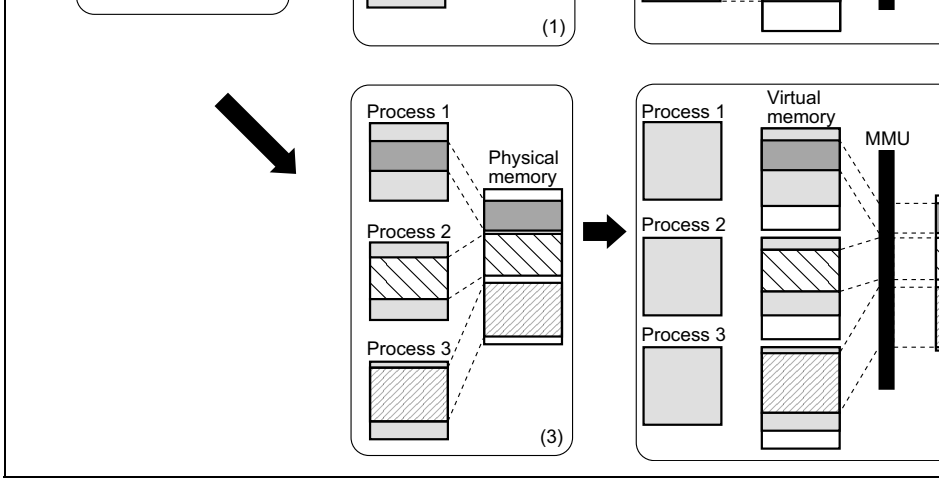


Figure 3.1 MMU Functions

(CCR) setting.

Mapping of the P1 area is fixed in physical address space (H'00000000 to H'1FFFF
the P1 area, setting a virtual address MSB (bit 31) to 0 generates the corresponding
address. P1 area accesses can be cached, and the cache control register (CCR) is se
whether to cache or not. Write-back or write-through mode can be selected.

Mapping of the P2 area is fixed in physical address space (H'00000000 to H'1FFFF
P2 area, setting the top three virtual address bits (bits 31, 30, and 29) to 0 generates
corresponding physical address. P2 area access cannot be cached.

The P1 and P2 areas are not mapped by the address translation table, so the TLB is
and no exceptions such as TLB misses occur. Initialization of MMU control registe
exception handling routines, and the like should be located in the P1 and P2 areas. F
that require high-speed processing should be placed in the P1 area, since it can be ca

Some peripheral module control registers are located in area 1 of the physical addre
When the physical address space is not used for address translation, these registers s
located in the P2 area. When address translation is to be used, set no caching.

The P4 area is used for mapping peripheral module register addresses, etc.

- User Mode

  In user mode, 2 Gbytes of the virtual address space from H'00000000 to H'7FFFFF
  U0) can be accessed. U0 is mapped onto physical address space in page units, in acc
  with address translation table information.

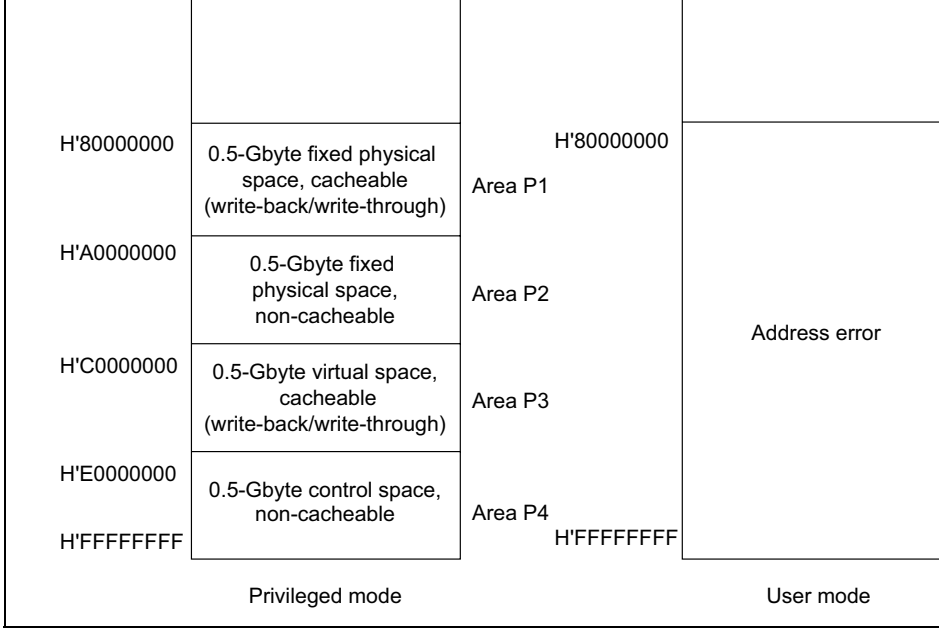


Figure 3.2 Virtual Address Space Mapping

Physical Address Space: The SH7709S supports a 32-bit physical address space, but bits are actually ignored and treated as a shadow. See section 10, Bus State Controller details.

Address Translation: When the MMU is enabled, the virtual address space is divided called pages. Physical addresses are translated in page units. Address translation tables memory hold information such as the physical address that corresponds to the virtual a memory protection codes. When an access to an area other than P4 occurs, if the acces address belongs to area P1 or P2 there is no TLB access and the physical address is un defined. If it belongs to area P0, P3, or U0, the TLB is searched by virtual address and virtual address is registered in the TLB, the access hits the TLB. The corresponding ph address and the page control information are read from the TLB and the physical addr determined.

SH7709S supports a 29-bit address space as the physical address space, the top 3 bits of physical address are ignored, and constitute a shadow space (see section 10, Bus State C (BSC)). For example, addresses H'00001000 in the P0 area, H'80001000 in the P1 area, H'A0001000 in the P2 area, and H'C0001000 in the P3 area are all mapped onto the sam address. When access to these addresses is performed with the cache enabled, an addres top 3 bits of the physical address masked to 0 is stored in the cache address array to ens congruity.

Single Virtual Memory Mode and Multiple Virtual Memory Mode: There are two v memory modes: single virtual memory mode and multiple virtual memory mode. In sin memory mode, multiple processes run in parallel using the virtual address space exclusi the physical address corresponding to a given virtual address is specified uniquely. In m virtual memory mode, multiple processes run in parallel sharing the virtual address spac given virtual address may be translated into different physical addresses depending on t Single or multiple virtual mode is selected by a value set in the MMU control register (In terms of operation, the only difference between single virtual memory mode and mul virtual memory mode is in the TLB address comparison method (see section 3.3.3, TLB Comparison).

Address Space Identifier (ASID): In multiple virtual memory mode, the address spac (ASID) is used to differentiate between processes running in parallel and sharing virtua space. The ASID is 8 bits in length and can be set by software setting of the ASID of th running process in the page table entry register high (PTEH) within the MMU. When th is switched using the ASID, the TLB does not have to be purged.

In single virtual memory mode, the ASID is used to provide memory protection for pro running simultaneously and using the virtual address space exclusively (see section 3.4 Software Management).

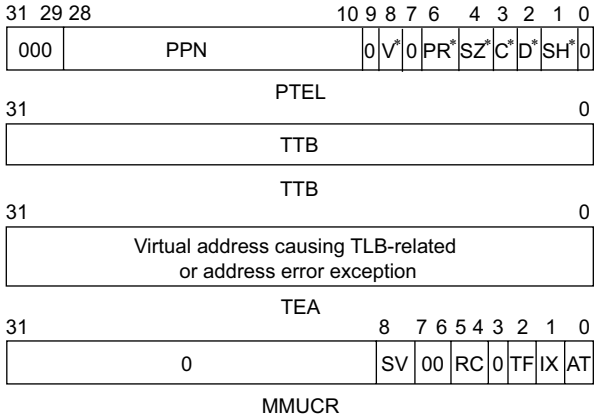
Page table entry register low	PTEL	R/W	Longword	Undefined	H
Translation table base register	TTB	R/W	Longword	Undefined	H
TLB exception address register	TEA	R/W	Longword	Undefined	H
MMU control register	MMUCR	R/W	Longword	*2	H

Notes: 1. Initialized by a power-on reset or manual reset.
2. SV bit: undefined
Other bits: 0

3.2 Register Description

There are five registers for MMU processing. These registers are located in address sp
and can only be accessed from privileged mode by specifying the address.

1. The page table entry register high (PTEH) register residing at address H'FFFFFFF
consists of a virtual page number (VPN) and ASID. The VPN set is the VPN of the
address at which the exception is generated in case of an MMU exception or addre
exception. When the page size is 4 kbytes, the VPN is the upper 20 bits of the virtu
but in this case the upper 22 bits of the virtual address are set. The VPN can also b
by software. As the ASID, software sets the number of the currently executing pro
VPN and ASID are recorded in the TLB by the LDTLB instruction.
2. The page table entry register low (PTEL) register residing at address H'FFFFFFF4
store the physical page number and page management information to be recorded
by the LDTLB instruction. The contents of this register are only modified in respo
software command. (Refer to section 3.4.3, MMU Instruction (LDTLB), and secti
MMU Exceptions.)
3. The translation table base register (TTB) residing at address H'FFFFFFF8, which p
base address of the current page table. The hardware does not set any value in TTB
automatically. TTB is available to software for general purposes.
4. The TLB exception address register (TEA) residing at address H'FFFFFFFC, whic
virtual address corresponding to a TLB or address error exception. This value rema
until the next exception or interrupt.



0: Reserved bits. Always read as 0. Writing is ignored. However, 0 should also be specified in a write to MMUCR only.

SV: Single virtual memory mode bit. 0: Multiple virtual memory mode
1: Single virtual memory mode

RC: A 2-bit random counter, automatically updated by hardware according to the following rules in the event of an MMU exception. When a TLB miss exception occurs, all TLB entry ways corresponding to the virtual address at which the exception occurred are checked, and if all ways are valid, 1 is added to RC; if there is one or more invalid way, they are set by priority from way 0, in the order: way 0, way 1, way 2, way 3. In the event of an MMU exception other than a TLB miss exception, the way which caused the exception is set in RC.

TF: TLB flush bit. Write 1 to flush the TLB (clear all valid bits of the TLB to 0). Always reads 0.

IX: Index mode bit. When 0, VPN bits 16–12 are used as the TLB index number. When 1, the value obtained by EX-ORing ASID bits 4–0 in PTEH and VPN bits 16–12 are used as the TLB index number.

AT: Address translation bit. Enables/disables the MMU.
0: MMU disabled
1: MMU enabled

Note: * Refer to section 3.3, TLB Functions.

Figure 3.3 MMU Register Contents

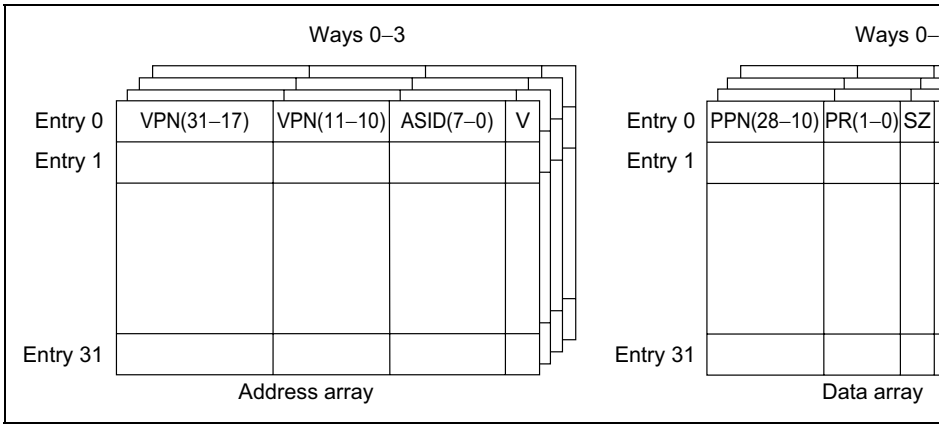
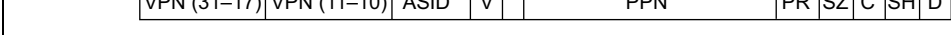


Figure 3.4 Overall Configuration of the TLB



TLB entry

Legend:

VPN: Virtual page number. Top 22 bits of virtual address for a 1-kbyte page, or top 20 bits of virtual address for a 4-kbyte page. Since VPN bits 16-12 are used as the index number, they are not stored in the TLB entry.

ASID: Address space identifier. Indicates the process that can access a virtual page. In single virtual memory mode and user mode, or in multiple virtual memory mode, if the SH bit is 0, the address is compared with the ASID in PTEH when address comparison is performed.

SH: Share status bit
0 = Page not shared between processes
1 = Page shared between processes

SZ: Page-size bit
0 = 1-kbyte page
1 = 4-kbyte page

V: Valid bit. Indicates whether entry is valid.
0 = Invalid
1 = Valid
Cleared to 0 by a power-on reset. Not affected by a manual reset.

PPN: Physical page number. Top 29 bits of physical address. PPN bits 11-10 are not used in case of a 4-kbyte page. Attention must be paid to the synonym problem in case of a 1-kbyte page (see section 3.4.4, Avoiding Synonym Problems).

PR: Set the most significant bit to 0.
Protection key field. 2-bit field encoded to define the access rights to the page.
00: Reading only is possible in privileged mode.
01: Reading/writing is possible in privileged mode.
10: Reading only is possible in privileged/user mode.
11: Reading/writing is possible in privileged/user mode.

C: Cacheable bit. Indicates whether the page is cacheable.
0: Non-cacheable
1: Cacheable

D: Dirty bit. Indicates whether the page has been written to.
0 = Not written to
1 = Written to

Figure 3.5 Virtual Address and TLB Structure

number

The second method is used to prevent lowered TLB efficiency that results when multi processes run simultaneously in the same virtual address space (multiple virtual memo specific entry is selected by generating an index number for each process. Figures 3.6 show the indexing schemes.

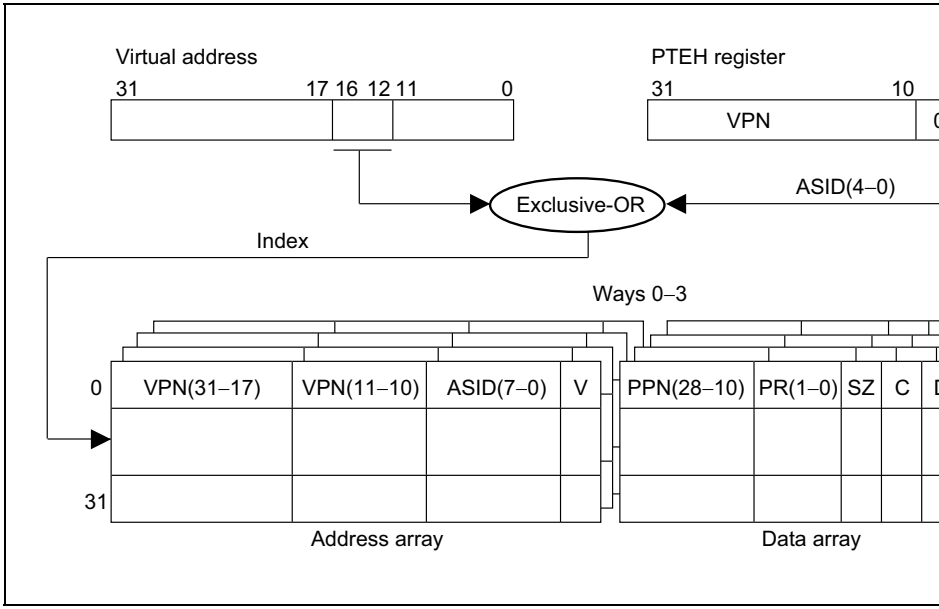


Figure 3.6 TLB Indexing (IX = 1)

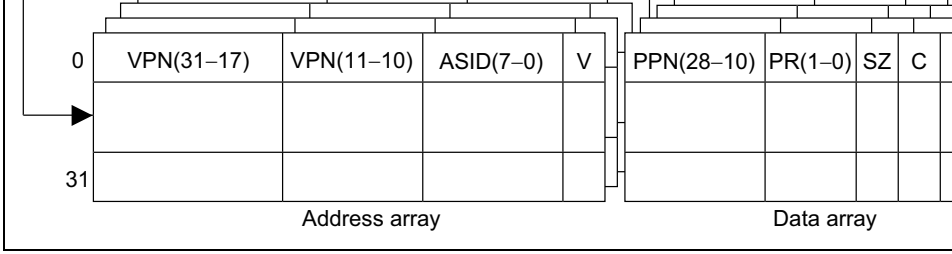


Figure 3.7 TLB Indexing (IX = 0)

3.3.3 TLB Address Comparison

The results of address comparison determine whether a specific virtual page number is in the TLB. The virtual page number of the virtual address that accesses external memo compared to the virtual page number of the indexed TLB entry. The ASID within the P compared to the ASID of the indexed TLB entry. All four ways are searched simultane compared values match, and the indexed TLB entry is valid (V bit = 1), the hit is regist

It is necessary to have software ensure that TLB hits do not occur simultaneously in mo way, as hardware operation is not guaranteed if this occurs. For example, if there are tw TLB entries with the same VPN and a setting is made such that a TLB hit is made only process with ASID = H'FF when one is in the shared state (SH = 1) and the other in the state (SH = 0), then if the ASID in PTEH is set to H'FF, there is a possibility of simulta hits in both these ways. It is therefore necessary to ensure that this kind of setting is not software.

The object compared varies depending on the page management information (SZ, SH) entry. It also varies depending on whether the system supports multiple virtual memory virtual memory.

The page-size information determines whether VPN (11–10) is compared. VPN (11–10 compared for 1-kbyte pages (SZ = 0) but not for 4-kbyte pages (SZ = 1).

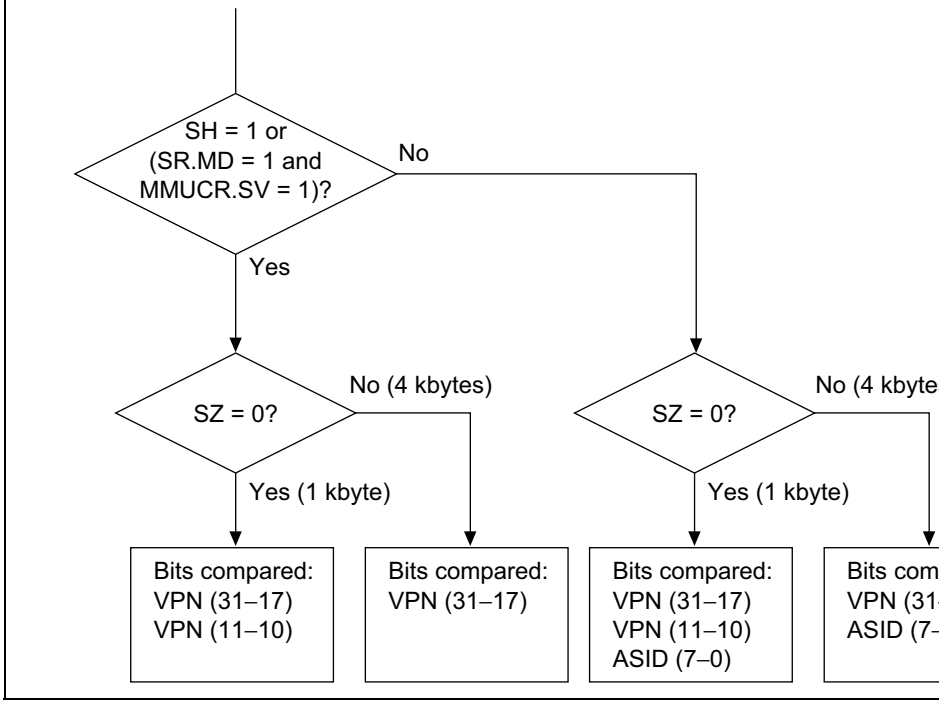


Figure 3.8 Objects of Address Comparison

memory.

The C bit in the entry indicates whether the referenced page resides in a cacheable or no
cacheable area of memory. When the control register in area 1 is mapped, set the C bit t
PR field specifies the access rights for the page in privileged and user modes and is use
memory. Attempts at nonpermitted accesses result in TLB protection violation exceptio

Access states designated by the D, C, and PR bits are shown in table 3.2.

Table 3.2 Access States Designated by D, C, and PR Bits

		Privileged Mode		User Mode	
		Reading	**Writing**	**Reading**	**Writing**
D bit	0	Permitted	Initial page write exception	Permitted	Initial pa exceptio
	1	Permitted	Permitted	Permitted	Permitte
C bit	0	Permitted (no caching)	Permitted (no caching)	Permitted (no caching)	Permitte (no cach
	1	Permitted (with caching)	Permitted (with caching)	Permitted (with caching)	Permitte (with cac
PR bit	00	Permitted	TLB protection violation exception	TLB protection violation exception	TLB prot violation
	01	Permitted	Permitted	TLB protection violation exception	TLB prot violation
	10	Permitted	TLB protection violation exception	Permitted	TLB prot violation
	11	Permitted	Permitted	Permitted	Permitte

determines the MMU exception and whether the cache is to be accessed (using the details of the determination method and the hardware processing, see section 3.5, Exceptions.

3.4.2 MMU Software Management

There are three kinds of MMU software management, as follows.

1. MMU register setting. MMUCR setting, in particular, should be performed in area for which address translation is not performed. Also, since SV and IX bit changes address translation system changes, in this case, TLB flushing should be performed simultaneously writing 1 to the TF bit also. Since MMU exceptions are not genera MMU disabled state with the AT bit cleared to 0, use in the disabled state must be with software that does not use the MMU.
2. TLB entry recording, deletion, and reading. TLB entry recording can be done in tw using the LDTLB instruction, or by writing directly to the memory-mapped TLB. entry deletion and reading, the memory allocation TLB can be accessed. See sectio MMU Instruction (LDTLB), for details of the LDTLB instruction, and section 3.6, Configuration of Memory-Mapped TLB, for details of the memory-mapped TLB.
3. MMU exception processing. When an MMU exception is generated, it is handled of information set from the hardware side. See section 3.5, MMU Exceptions, for

When single virtual memory mode is used, it is possible to create a state in which phys memory access is enabled in the privileged mode only by clearing the share status bit specify recording of all TLB entries. This strengthens inter-process memory protection enables special access levels to be created in the privileged mode only.

Recording a 1-kbyte page TLB entry may result in a synonym problem. See section 3. Avoiding Synonym Problems.

When an MMU exception occurs, the virtual page number of the virtual address that ca
exception is set in PTEH by hardware. The way is set in the RC bit of MMUCR for eac
exception (see figure 3.3). Consequently, if the LDTLB instruction is issued after settin
PTEL in the MMU exception handling routine, TLB entry recording is possible. Any T
can be updated by software rewriting of PTEH and the RC bits in MMUCR.

As the LDTLB instruction changes address translation information, there is a risk of de
address translation information if this instruction is issued in the P0, U0, or P3 area. Ma
therefore, that this instruction is issued in the P1 or P2 area. Also, an instruction associa
access to the P0, U0, or P3 area (such as the RTE instruction) should be issued at least t
instructions after the LDTLB instruction.

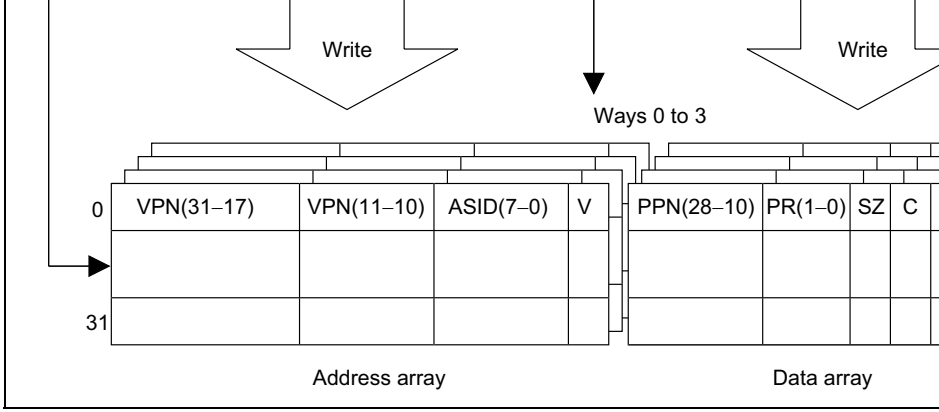


Figure 3.9 Operation of LDTLB Instruction

address bits 11–4. When a 4-kbyte page is used, virtual address bits 11–4 are included i
offset, and since they are not subject to address translation, they are the same as physica
bits 11–4. In cache-based address comparison and recording in the address array, since
tag address is a physical address, physical address bits 28–10 are recorded.

When a 1-kbyte page is used, also, a cache index number is created using virtual addres
However, in case of a 1-kbyte page, virtual address bit (11, 10) is subject to address tra
and therefore may not be the same as physical address bit (11, 10). Consequently, the p
address is recorded in a different entry from that of the index number indicated by the p
address in the cache address array.

For example, assume that, with 1-kbyte page TLB entries, TLB entries for which the fo
translation has been performed are recorded in two TLBs:

Virtual address 1	H'00000000	→	physical address	H'00000400
Virtual address 2	H'00000400	→	physical address	H'00000400

Virtual address 1 is recorded in cache entry H'00, and virtual address 2 in cache entry H
two virtual addresses are recorded in different cache entries despite the fact that the phy
addresses are the same, memory inconsistency will occur as soon as a write is performe
virtual address. Therefore, when recording a 1-kbyte TLB entry, if the physical address
as a physical address already used in another TLB entry, it should be recorded in such a
physical address bit (11, 10) is the same.

Note: In readiness for the future expansion of the SuperH RISC engine family, we rec
that, when multiple sets of address translation information are mapped onto the
physical area of memory, you set the VPN numbers so that each VPN [20:10] is
the others. We also recommend that you do not map multiple sets of address-tra
information that include 1- and 4-kbyte pages to a single physical area.

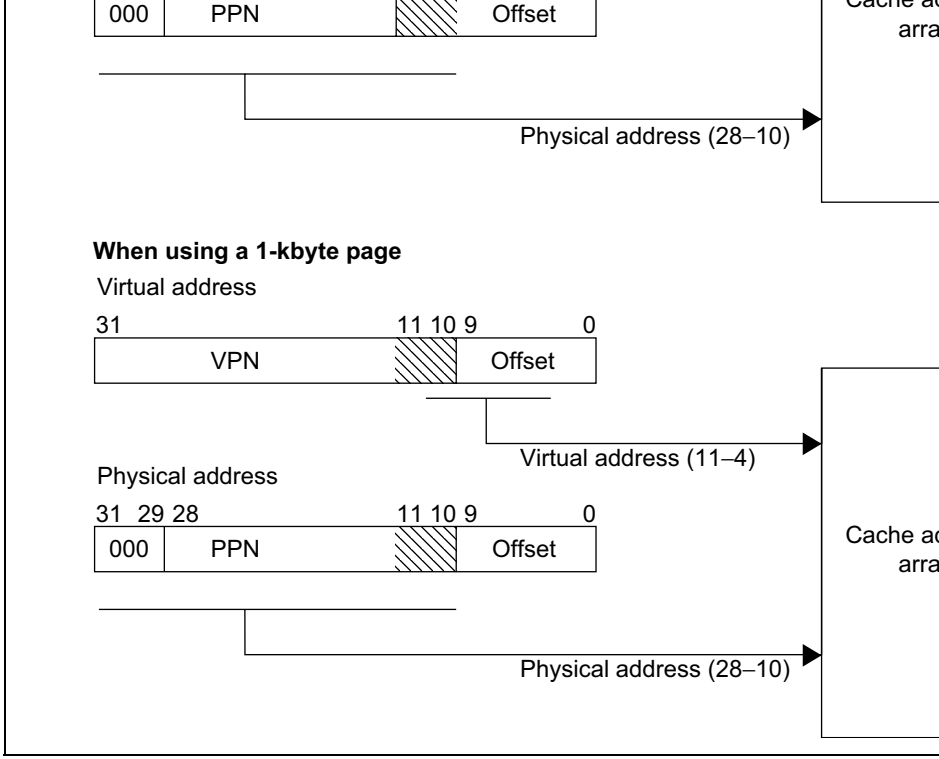


Figure 3.10 Synonym Problem

software operations.

Hardware Operations: In a TLB miss, the SH7709S hardware executes a set of prescr operations, as follows:

1. The VPN field of the virtual address causing the exception is written to the PTEH re
2. The virtual address causing the exception is written to the TEA register.
3. Either exception code H'040 for a load access, or H'060 for a store access, is written EXPEVT register.
4. The PC value indicating the address of the instruction in which the exception occurr written to the save program counter (SPC). If the exception occurred in a delay slot, value indicating the address of the related delayed branch instruction is written to th
5. The contents of the status register (SR) at the time of the exception are written to the status register (SSR).
6. The mode (MD) bit in SR is set to 1 to place the SH7709S in the privileged mode.
7. The block (BL) bit in SR is set to 1 to mask any further exception requests.
8. The register bank (RB) bit in SR is set to 1.
9. The random counter (RC) field in the MMU control register (MMUCR) is incremen when all ways are checked for the TLB entry corresponding to the logical address a exception occurred, and all ways are valid. If one or more ways are invalid, those w in RC in prioritized order from way 0 through way 1, way 2, and way 3.
10. Execution branches to the address obtained by adding the value of the VBR content H'00000400 to invoke the user-written TLB miss exception handler.

Software (TLB Miss Handler) Operations: The software searches the page tables in e memory and allocates the required page table entry. Upon retrieving the required page t software must execute the following operations:

1. Write the value of the physical page number (PPN) field and the protection key (PR (SZ), cacheable (C), dirty (D), share status (SH), and valid (V) bits of the page table recorded in the address translation table in the external memory into the PTEL regis SH7709S.

A TLB protection violation exception results when the virtual address and the address selected TLB entry are compared and a valid entry is found to match, but the type of a permitted by the access rights specified in the PR field. TLB protection violation exce processing includes both hardware and software operations.

Hardware Operations: In a TLB protection violation exception, the SH7709S hardw a set of prescribed operations, as follows:

1. The VPN field of the virtual address causing the exception is written to the PTEH
2. The virtual address causing the exception is written to the TEA register.
3. Either exception code H'0A0 for a load access, or H'0C0 for a store access, is writt EXPEVT register.
4. The PC value indicating the address of the instruction in which the exception occu written into SPC (if the exception occurred in a delay slot, the PC value indicating of the related delayed branch instruction is written into SPC).
5. The contents of SR at the time of the exception are written to SSR.
6. The MD bit in SR is set to 1 to place the SH7709S in the privileged mode.
7. The BL bit in SR is set to 1 to mask any further exception requests.
8. The register bank (RB) bit in SR is set to 1.
9. The way that generated the exception is set in the RC field in MMUCR.
10. Execution branches to the address obtained by adding the value of the VBR conten H'00000100 to invoke the TLB protection violation exception handler.

Software (TLB Protection Violation Handler) Operations: Software resolves the T protection violation and issues the RTE (return from exception handler) instruction to the handler and return to the instruction stream.

1. The VPN number of the virtual address causing the exception is written to the PTEH
2. The virtual address causing the exception is written to the TEA register.
3. The way number causing the exception is written to RC in MMUCR.
4. Either exception code H'040 for a load access, or H'060 for a store access, is written EXPEVT register.
5. The PC value indicating the address of the instruction in which the exception occurr written to the SPC. If the exception occurred in a delay slot, the PC value indicating address of the delayed branch instruction is written to the SPC.
6. The contents of SR at the time of the exception are written into SSR.
7. The mode (MD) bit in SR is set to 1 to place the SH7709S in the privileged mode.
8. The block (BL) bit in SR is set to 1 to mask any further exception requests.
9. The register bank (RB) bit in SR is set to 1.
10. Execution branches to the address obtained by adding the value of the VBR content H'00000100, and the TLB protection violation exception handler starts.

Software (TLB Invalid Exception Handler) Operations: The software searches the p in external memory and assigns the required page table entry. Upon retrieving the requi table entry, software must execute the following operations:

1. Write the values of the physical page number (PPN) field and the values of the prot (PR), page size (SZ), cacheable (C), dirty (D), share status (SH), and valid (V) bits table entry recorded in the external memory to the PTEL register.
2. If using software for way selection for entry replacement, write the desired value to field in MMUCR.
3. Issue the LDTLB instruction to load the contents of PTEH and PTEL into the TLB.
4. Issue the RTE instruction to terminate the handler and return to the instruction strea RTE instruction should be issued after two LDTLB instructions.

1. The VPN field of the virtual address causing the exception is written to the PTEH
2. The virtual address causing the exception is written to the TEA register.
3. Exception code H'080 is written to the EXPEVT register.
4. The PC value indicating the address of the instruction in which the exception occu written to the SPC. If the exception occurred in a delay slot, the PC value indicatin address of the related delayed branch instruction is written to the SPC.
5. The contents of SR at the time of the exception are written to SSR.
6. The MD bit in SR is set to 1 to place the SH7709S in the privileged mode.
7. The BL bit in SR is set to 1 to mask any further exception requests.
8. The register bank (RB) bit in SR is set to 1.
9. The way that caused the exception is set in the RC field in MMUCR.
10. Execution branches to the address obtained by adding the value of the VBR conten H'00000100 to invoke the user-written initial page write exception handler.

Software (Initial Page Write Handler) Operations: The software must execute the operations:

1. Retrieve the required page table entry from external memory.
2. Set the D bit of the page table entry in the external memory to 1.
3. Write the value of the PPN field and the PR, SZ, C, D, SH, and V bits of the page in the external memory to the PTEL register.
4. If using software for way selection for entry replacement, write the desired value t field in MMUCR.
5. Issue the LDTLB instruction to load the contents of PTEH and PTEL into the TLB
6. Issue the RTE instruction to terminate the handler and return to the instruction stre RTE instruction should be issued after two LDTLB instructions.

Figure 3.11 shows the flowchart for MMU exceptions.

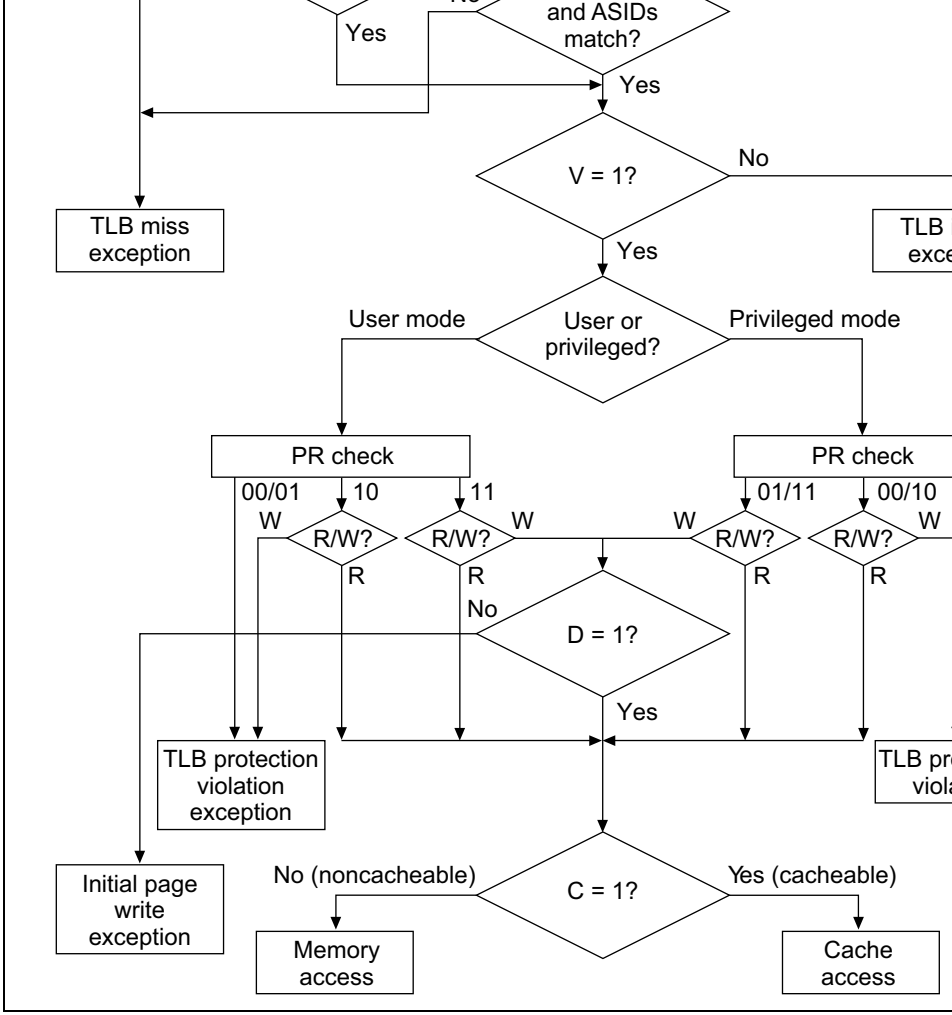


Figure 3.11 MMU Exception Generation Flowchart

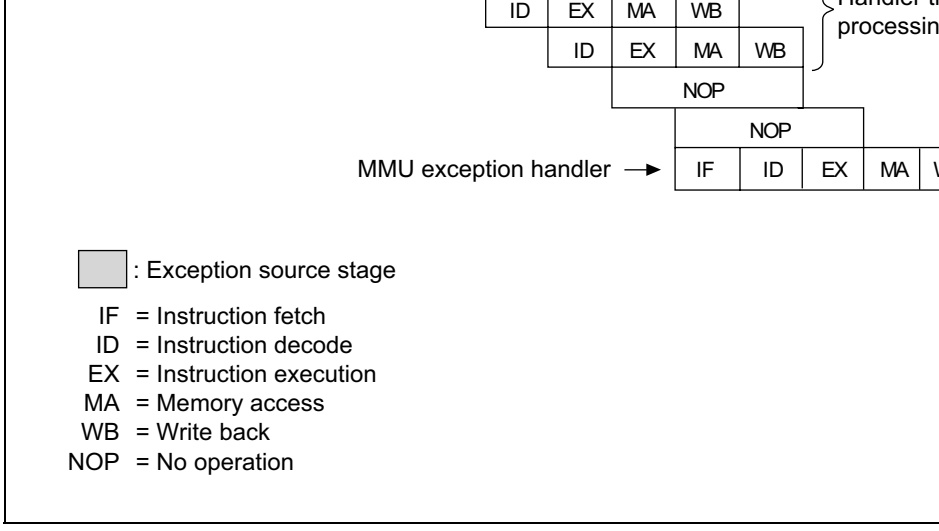


Figure 3.12 MMU Exception Signals in Instruction Fetch

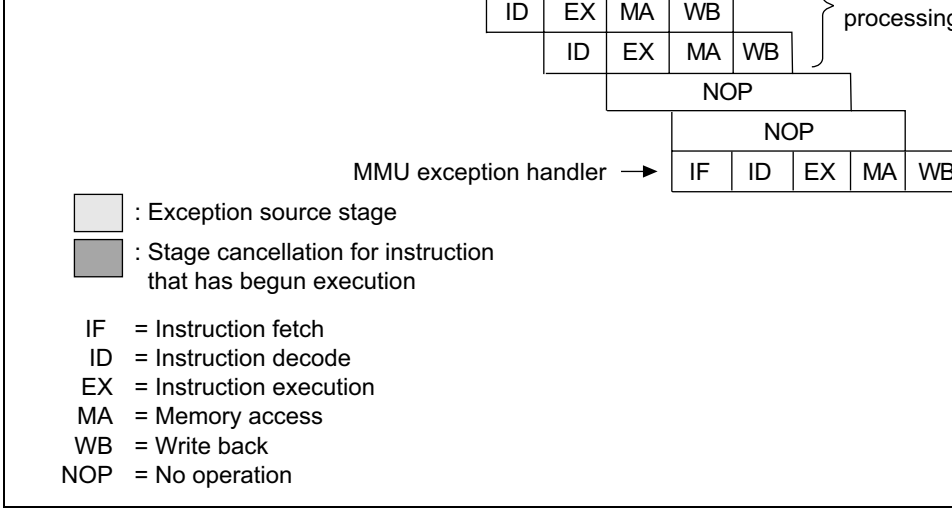


Figure 3.13 MMU Exception Signals in Data Access

3.6 Configuration of Memory-Mapped TLB

To allow the management of TLB operations by software, the MOV instruction can be
privileged mode, to read and write TLB contents. The TLB is mapped to the P4 area of
address space. The TLB address array (VPN, V bit, and ASID) is mapped to H'F20000
H'F2FFFFFF, and the TLB data array (PPN, PR, SZ, CD, S, and H bits) is mapped to H
to H'F3FFFFFF. It is also possible to access the V bits in the address array from the dat
Only longword access is possible, for both the address and data arrays.

3.6.1 Address Array

The address array is mapped to H'F2000000 to H'F2FFFFFF. To access the address arra
bit address field (for read/write access) and 32-bit data field (for write access) must be s
The address field has the information that selects the entry to be accessed; the data field
the VPN, the V bit, and the ASID to be written to the address array (figure 3.14 (1)).

(2) Address Array Write

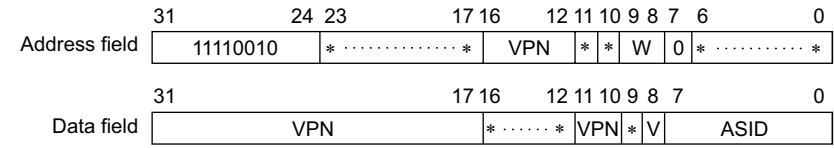
Writes the data set in the data field to the entry that corresponds to the entry addres
that were specified in the address field.

3.6.2 Data Array

The data array is assigned to H'F3000000 to H'F3FFFFFF. To access a data array, the
address field (for read/write operations), and 32-bit data field (for write operations) m
specified. These are specified in the general register. The address section specifies info
selecting the entry to be accessed; the data section specifies the longword data to be w
data array (figure 3.14 (2)).

In the address section, specify the entry address for selecting the entry (bits 16–12), W
selecting the way (bits 9–8: 00 is way 0, 01 is way 1, 10 is way 2, 11 is way 3), and H
indicate data array access (bits 31–24). The IX bit in MMUCR indicates whether an E
taken of the entry address and ASID.

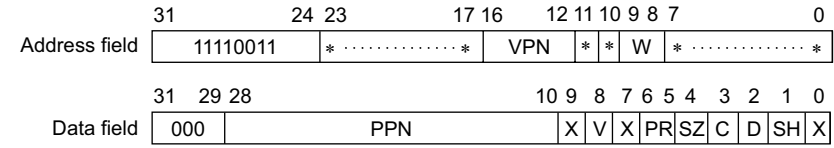
Both reading and writing use the longword of the data array specified by the index add
way number. The access size of the data array is fixed at longword.



VPN: Virtual page number ASID: Address space identif
V: Valid bit *: Don't care bit
W: Way (00: Way 0, 01: Way 1, 10: Way 2, 11: Way 3)

(2) TLB Data Array Access

Read/write access



PPN: Physical page number V: Valid bit
PR: Protection key field SZ: Page-size bit
C: Cacheable bit D: Dirty bit
SH: Share status bit *: Don't care bit
VPN: Virtual page number
X: 0 for read, don't care bit for write
W: Way (00: Way 0, 01: Way 1, 10: Way 2, 11: Way 3)

Figure 3.14 Specifying Address and Data for Memory-Mapped TLB Acc

```
; MMUCR.IX=0
; VPN(31–17)=B'0001 0101 0100 011   VPN(11–10)=B'10   ASID=B'0
; corresponding entry association is made from the entry selec
; the VPN(16–12)=B'1 0011 index, the V bit of the hit way is c
; 0,achieving invalidation.
MOV.L  R0,@R1
```

Reading the Data of a Specific Entry: This example reads the data section of a speci entry. The bit order indicated in the data field in figure 3.14 (2) is read. R0 specifies th and the data section of a selected entry is read to R1.

```
; R1=H'F300 4300   VPN(16-12)=B'00100   Way 3
MOV.L @R0,R1
```

3.7 Usage Note

The operations listed below must only be performed when the TLB is disabled or in th area. Any subsequent operation that accesses the P0, P3, or U0 area must take place tw instructions after any of the below operations.

1. Change SR.MD or SR.BL
2. Execute the LDTLB instruction
3. Write to the memory-mapped TLB
4. Change MMUCR.

termination of the executing instruction, control is passed to a user-written exception h
However, in response to an interrupt request, normal program execution continues unt
the executing instruction. Here, all exceptions other than resets and interrupts will be
general exceptions. There are thus three types of exceptions: resets, general exception
interrupts.

4.1.2 Register Configuration

Table 4.1 lists the registers used for exception handling. A register with an undefined
should be initialized by software.

Table 4.1 Register Configuration

Register	Abbr.	R/W	Size	Initial Value	Ad
TRAPA exception register	TRA	R/W	Longword	Undefined	H'
Exception event register	EXPEVT	R/W	Longword	Power-on reset: H'000 Manual reset: H'020*1	H'
Interrupt event register	INTEVT	R/W	Longword	Undefined	H'
Interrupt event register2	INTEVT2	R	Longword	Undefined	H' (H

Notes: 1. H'000 is set in a power-on reset, and H'020 in a manual reset.
2. When address translation by the MMU does not apply, the address in parer
should be used.

4.2 Exception Handling Function

4.2.1 Exception Handling Flow

In exception handling, the contents of the program counter (PC) and status register (SI
in the saved program counter (SPC) and saved status register (SSR), respectively, and
the exception handler is invoked from a vector address. The return from exception han
instruction is issued by the exception handler routine on completion of the routine, res

5. An exception code identifying the exception event is written to bits 11–0 of the exce event (EXPEVT) or interrupt event (INTEVT or INTEVT2) register.
6. Instruction execution jumps to the designated exception vector address to invoke th routine.

4.2.2 Exception Vector Addresses

The reset vector address is fixed at H'A0000000. The other three events are assigned of the vector base address by software. Translation look-aside buffer (TLB) miss exceptio offset from the vector base address of H'00000400. The vector address offset for genera events other than TLB miss exceptions is H'00000100. The interrupt vector address offs H'00000600. The vector base address is loaded into the vector base register (VBR) by s The vector base address should reside in P1 or P2 fixed physical address space. Figure 4 the relationship between the vector base address, the vector offset, and the vector table.

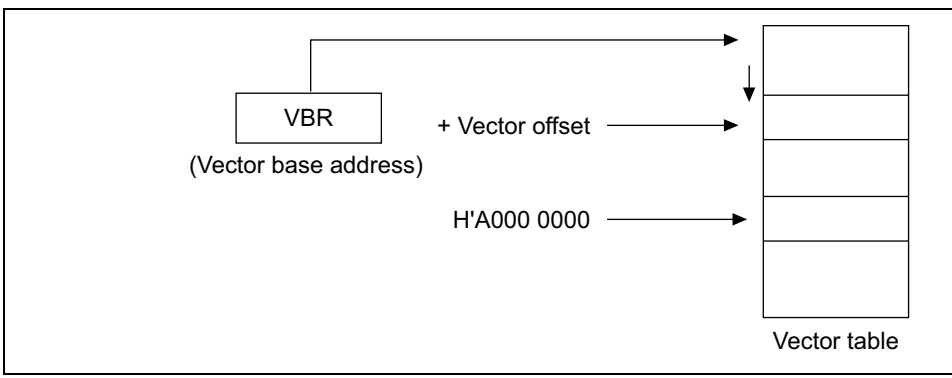


Figure 4.1 Vector Table

In table 4.2, exceptions and their vector addresses are listed by exception type, instructi completion state, relative acceptance priority, relative order of occurrence within an ins execution sequence and vector address for exceptions and their vector addresses.

events		TLB miss	2	2	—
		TLB invalid (instruction access)	2	3	—
		TLB protection violation (instruction access)	2	4	—
		General illegal instruction exception	2	5	—
		Illegal slot instruction exception	2	5	—
		CPU address error (data access)	2	6	—
		TLB miss (data access not in repeat loop)	2	7	—
		TLB invalid (data access)	2	8	—
		TLB protection violation (data access)	2	9	—
		Initial page write	2	10	—
	Completed	Unconditional trap (TRAPA instruction)	2	5	—
		User breakpoint trap	2	n[*2]	—
		DMA address error	2	—	—
General interrupt requests	Completed	Nonmaskable interrupt	3	—	—
		External hardware interrupt	4[*3]	—	—
		UDI interrupt	4[*3]	—	—

Notes: 1. Priorities are indicated from high to low, 1 being the highest and 4 the lowe

2. The user defines the break point traps. 1 is a break point before instruction and 11 is a break point after instruction execution. For an operand break po

3. Use software to specify relative priorities of external hardware interrupts an module interrupts (see section 6, Interrupt Controller (INTC)).

detected in a subsequent instruction.

Three general exception events (reserved instruction code exception, unconditional trap illegal instruction exception) are detected in the decode stage (ID stage) of different ins and are mutually exclusive events in the instruction pipeline. They have the same execu priority. Figure 4.2 shows the order of general exception acceptance.

RIE (reserved instruction exception)

Detection Order:

TLB miss (instruction n+1)

↓

TLB miss (instruction n) and general illegal instruction exception (instruction n
= simultaneous detection

Handling Order:	**Program Order:**
TLB miss (instruction n)	
↓	1
Re-execution of instruction n	
↓	
TLB miss (instruction n + 1)	
↓	2
Re-execution of instruction n + 1	
↓	
RIE (instruction n + 2)	3

IF = Instruction fetch
ID = Instruction decode
EX = Instruction execution
MA = Memory access
WB = Write back

Figure 4.2 Example of Acceptance Order of General Exceptions

All exceptions other than a reset are detected in the pipeline ID stage, and accepted at
boundaries. However, an exception is not accepted between a delayed branch instructi
delay slot. A re-execution type exception detected in a delay slot is accepted before ex
the delayed branch instruction. A completion type exception detected in a delayed bra

Table 4.3 Exception Codes

Exception Type	Exception Event	Excepti
Reset	Power-on reset	H'000
	Manual reset	H'020
	UDI reset	H'000
General exception events	TLB miss/invalid (read)	H'040
	TLB miss/invalid (write)	H'060
	Initial page write	H'080
	TLB protection violation (read)	H'0A0
	TLB protection violation (write)	H'0C0
	CPU address error (read)	H'0E0
	CPU address error (write)	H'100
	Unconditional trap (TRAPA instruction)	H'160
	Illegal general instruction exception	H'180
	Illegal slot instruction exception	H'1A0
	User breakpoint trap	H'1E0
	DMA address error	H'5C0
General interrupt requests	Nonmaskable interrupt	H'1C0
	UDI interrupt	H'5E0
	External hardware interrupts:	
	IRL3–IRL0 = 0000	H'200
	IRL3–IRL0 = 0001	H'220

IRL3–IRL0 = 1000	H'300
IRL3–IRL0 = 1001	H'320
IRL3–IRL0 = 1010	H'340
IRL3–IRL0 = 1011	H'360
IRL3–IRL0 = 1100	H'380
IRL3–IRL0 = 1101	H'3A0
IRL3–IRL0 = 1110	H'3C0

4.2.5 Exception Request Masks

When the BL bit in SR is 0, exceptions and interrupts are accepted.

If a general exception event occurs when the BL bit in SR is 1, the CPU's internal regi to their post-reset state, other module registers retain their contents prior to the genera and a branch is made to the same address (H'A0000000) as for a reset.

If a general interrupt occurs when BL = 1, the request is masked (held pending) and n until the BL bit is cleared to 0 by software. For reentrant exception handling, SPC and be saved and the BL bit in SR cleared to 0.

4.2.6 Returning from Exception Handling

The RTE instruction is used to return from exception handling. When RTE is executed value is set in PC, and the SSR value in SR, and the return from exception handling is by branching to the SPC address.

If SPC and SSR have been saved in external memory, set the BL bit in SR to 1, then re and SSR, and issue an RTE instruction.

EXPEVT can also be modified by software.

2. The interrupt event register (INTEVT) resides at address H'FFFFFFD8, and contain interrupt exception code or a code indicating the interrupt priority. Which is set whe interrupt occurs depends on the interrupt source (see tables 6.4 and 6.5). The except interrupt priority code is set automatically by hardware when an exception occurs. I can also be modified by software.
3. Interrupt event register 2 (INTEVT2) resides at address H'04000000, and contains a exception code. The exception code set in INTEVT2 is that for an interrupt request. exception code is set automatically by hardware when an exception occurs.
4. The TRAPA exception register (TRA) resides at address H'FFFFFFD0, and contain immediate data (imm) for the TRAPA instruction. TRA is set automatically by hard a TRAPA instruction is executed. TRA can also be modified by software.

The bit configurations of the EXPEVT, INTEVT, INTEVT2, and TRA registers are sho figure 4.3.

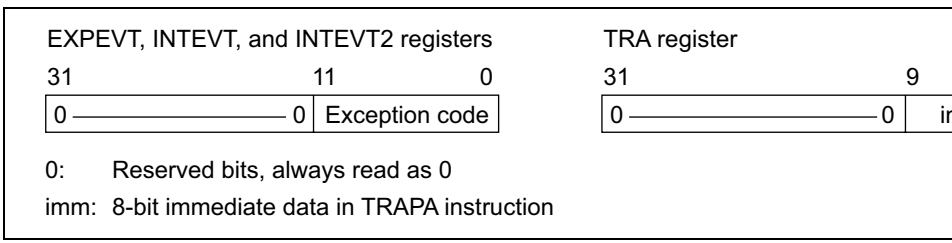


Figure 4.3 Bit Configurations of EXPEVT, INTEVT, INTEVT2, and TRA Re

sequence consists of the following operations:

1. The MD bit in SR is set to 1 to place the SH7709S in privileged mode.
2. The BL bit in SR is set to 1, masking any subsequent exceptions (except the NMI i when the BLMSK bit is 1).
3. The RB bit in SR is set to 1.
4. An encoded value of H'000 in a power-on reset or H'020 in a manual reset is writte 0 of the EXPEVT register to identify the exception event.
5. Instruction execution jumps to the user-written exception handler at address H'A0

4.4.2 Interrupts

An interrupt handling request is accepted on completion of the current instruction. The acceptance sequence consists of the following operations:

1. The contents of PC and SR are saved to SPC and SSR, respectively.
2. The BL bit in SR is set to 1, masking any subsequent exceptions (except the NMI i when the BLMSK bit is 1).
3. The MD bit in SR is set to 1 to place the SH7709S in privileged mode.
4. The RB bit in SR is set to 1.
5. An encoded value identifying the exception event is written to bits 11–0 of the IN INTEVT2 registers.
6. Instruction execution jumps to the vector location designated by the sum of the val contents of the vector base register (VBR) and H'00000600 to invoke the exceptio

4. The RB bit in SR is set to 1.
5. Instruction execution jumps to the vector location designated by either the sum of th base address and offset H'00000400 in the vector table in a TLB miss trap, or by the vector base address and offset H'00000100 for exceptions other than TLB miss traps the exception handler.

4.5 Individual Exception Operations

This section describes the conditions for specific exception handling, and this LSI oper

4.5.1 Resets

- Power-On Reset
  - Conditions: $\overline{\text{RESETP}}$ low
  - Operations: EXPEVT set to H'000, VBR and SR initialized, branch to PC = H'A Initialization sets the VBR register to H'0000000. In SR, the MD, RB and BL bi 1 and the interrupt mask bits (I3 to I0) are set to B'1111. The CPU and on-chip p modules are initialized. See the register descriptions in the relevant sections for power-on reset must always be performed when powering on. A low level is out the $\overline{\text{RESETOUT}}$ pin, and a high level is output from the STATUS0 and STATU

- Manual Reset
  - Conditions: $\overline{\text{RESETM}}$ low
  - Operations: EXPEVT set to H'020, VBR and SR initialized, branch to PC = H'A Initialization sets the VBR register to H'0000000. In SR, the MD, RB, and BL b to 1 and the interrupt mask bits (I3 to I0) are set to B'1111. The CPU and on-chi peripheral modules are initialized. See the register descriptions in the relevant se details. A low level is output from the $\overline{\text{RESETOUT}}$ pin, and a high level is outpu STATUS0 and STATUS1 pins.

Type	Conditions for Transition to Reset State	Internal State	
		CPU	On-Chip Peripher
Power-on reset	$\overline{\text{RESETP}}$ = Low	Initialized	(See register config relevant sections)
Manual reset	$\overline{\text{RESETM}}$ = Low	Initialized	
UDI reset	UDI reset command input	Initialized	

4.5.2 General Exceptions

- TLB miss exception
  - Conditions: Comparison of TLB addresses shows no address match.
  - Operations: The virtual address (32 bits) that caused the exception is set in TE corresponding virtual page number (22 bits) is set in PTEH (31–10). The ASI indicates the ASID at the time the exception occurred. If all ways are valid, 1 i the RC bit in MMUCR. If there is one or more invalid way, they are set by pri with way 0.

  PC and SR of the instruction that generated the exception are saved to SPC and SS respectively. If the exception occurred during a read, H'040 is set in EXPEVT; if t occurred during a write, H'060 is set in EXPEVT. The BL, MD and RB bits in SR and a branch occurs to PC = VBR + H'0400.

  To speed up TLB miss processing, the offset differs from other exceptions.

respectively. If the exception occurred during a read, H'040 is set in EXPEVT; if the occurred during a write, H'060 is set in EXPEVT. The BL, MD, and RB bits in SR a and a branch occurs to PC = VBR + H'0100.

- Initial page write exception
  - Conditions: A hit occurred to the TLB for a store access, but the TLB entry data 0.
    This occurs for initial writes to the page registered by the load.
  - Operations: The virtual address (32 bits) that caused the exception is set in TEA corresponding virtual page number (22 bits) is set in PTEH (31–10). The ASID indicates the ASID at the time the exception occurred. The way that generated th exception is set in the RC bit in MMUCR.

  PC and SR of the instruction that generated the exception are saved to SPC and SSR respectively. H'080 is set in EXPEVT. The BL, MD, and RB bits in SR are set to 1 branch occurs to PC = VBR + H'0100.

- TLB protection exception
  - Conditions: When a hit access violates the TLB protection information (PR bits) below:

PR	Privileged mode	User mode
00	Only read enabled	No access
01	Read/write enabled	No access
10	Only read enabled	Only read enabled
11	Read/write enabled	Read/write enabled

  - Operations: The virtual address (32 bits) that caused the exception is set in TEA corresponding virtual page number (22 bits) is set in PTEH (31–10). The ASID indicates the ASID at the time the exception occurred. The way that generated th exception is set in the RC bits in MMUCR.

  PC and SR of the instruction that generated the exception are saved to SPC and SSR respectively. If the exception occurred during a read, H'0A0 is set in EXPEVT; if the occurred during a write, H'0C0 is set in EXPEVT. The BL, MD, and RB bits in SR and a branch occurs to PC = VBR + H'0100.

SR of the instruction that generated the exception are saved to SPC and SSR, re
If the exception occurred during a read, H'0E0 is set in EXPEVT; if the excepti
during a write, H'100 is set in EXPEVT. The BL, MD, and RB bits in SR are s
branch occurs to PC = VBR + H'0100. See section 3.5.5, Processing Flow in E
MMU Exception, for more information.

- Unconditional trap
  - Conditions: TRAPA instruction executed
  - Operations: The exception is a processing-completion type, so PC of the instru
    the TRAPA instruction is saved to SPC. SR from the time when the TRAPA in
    was executing is saved to SSR. The 8-bit immediate value in the TRAPA instru
    quadrupled and set in TRA (9–0). H'160 is set in EXPEVT. The BL, MD, and
    SR are set to 1 and a branch occurs to PC = VBR + H'0100.

- Illegal general instruction exception
  - Conditions:
    a. When undefined code not in a delay slot is decoded

       Delay branch instructions: JMP, JSR, BRA, BRAF, BSR, BSRF, RTS, RTE
       BF/S

       Undefined instruction: H'Fxxx

    b. When a privileged instruction not in a delay slot is decoded in user mode

       Privileged instructions: LDC, STC, RTE, LDTLB, SLEEP; Instructions tha
       GBR with LDC/STC are not privileged instructions and therefore do not ap
  - Operations: PC and SR of the instruction that generated this instruction are sav
    and SSR, respectively. H'180 is set in EXPEVT. The BL, MD, and RB bits in S
    1 and a branch occurs to PC = VBR + H'100. When an undefined code other th
    decoded, operation cannot be guaranteed.

c. When a privileged instruction in a delay slot is decoded in user mode

Privileged instructions: LDC, STC, RTE, LDTLB, SLEEP; Instructions that GBR with LDC/STC are not privileged instructions and therefore do not app

— Operations: PC of the immediately preceding delay branch instruction is saved t of the instruction that generated the exception is saved to SSR. H'1A0 is set in E The BL, MD, and RB bits in SR are set to 1 and a branch occurs to PC = VBR + When an undefined instruction other than H'Fxxx is decoded, operation cannot b guaranteed.

- User break point trap
  - — Conditions: When a break condition set in the user break controller is satisfied
  - — Operations: When a post-execution break occurs, PC of the instruction immedia the instruction that set the break point is set in SPC. If a pre-execution break occ the instruction that set the break point is set in SPC. SR when the break occurs is SSR. H'1E0 is set in EXPEVT. The BL, MD, and RB bits in SR are set to 1 and occurs to PC = VBR + H'0100. See section 7, User Break Controller, for more in

- DMA address error
  - — Conditions:
    a. Word data accessed from addresses other than word boundaries (4n + 1, 4n +
    b. Longword accessed from addresses other than longword boundaries (4n + 1, 4n + 3)
  - — Operations: PC of the instruction immediately after the instruction executed befo exception occurs is saved to SPC. SR when the exception occurs is saved to SSR set in EXPEVT. The BL, MD, and RB bits in SR are set to 1 and a branch occur VBR + H'0100.

section 6, Interrupt Controller (INTC), for more information.

2. IRL Interrupts
   — Conditions: The value of the interrupt mask bits in SR is lower than the IRL3–I and the BL bit in SR is 0. The interrupt is accepted at an instruction boundary.
   — Operations: The PC value after the instruction at which the interrupt is accepte SPC. SR at the time the interrupt is accepted is saved to SSR. The code corresp the IRL3–IRL0 level is set in INTEVT and INTEVT2. The corresponding code H'200 + [IRL3–IRL0] × H'20. See table 6.5, for the corresponding codes. The RB bits in SR are set to 1 and a branch occurs to VBR + H'0600. The received set in SR.IMASK. See section 6, Interrupt Controller (INTC), for more inform

3. IRQ Pin Interrupts
   — Conditions: The IRQ pin is asserted, SR.IMASK is lower than the IRQ priority the BL bit in SR is 0. The interrupt is accepted at an instruction boundary.
   — Operations: The PC value after the instruction at which the interrupt is accepte SPC. SR at the point the interrupt is accepted is saved to SSR. The code corres the interrupt source is set in INTEVT and INTEVT2. The BL, MD, and RB b set to 1 and a branch occurs to VBR + H'0600. The received level is not set in t mask bits in SR. See section 6, Interrupt Controller (INTC), for more informati

4. PINT Pin Interrupts
   — Conditions: The PINT pin is asserted, the interrupt mask bits in SR. is lower th priority level, and the BL bit in SR is 0. The interrupt is accepted at an instruct boundary.
   — Operations: The PC value after the instruction at which the interrupt is accepte SPC. SR at the point the interrupt is accepted is saved to SSR. The code corres the interrupt source is set in INTEVT and INTEVT2. The BL, MD, and RB b set to 1 and a branch occurs to VBR + H'0600. The received level is not set in t mask bits in SR. See section 6, Interrupt Controller (INTC), for more informati

6. UDI Interrupt
   - Conditions: An UDI interrupt command is input (see section 22.4.4, UDI Interru SR.IMASK is lower than 15, and the BL bit in SR is 0. The interrupt is accepted instruction boundary.
   - Operations: The PC value after the instruction that accepts the interrupt is saved SR at the point the interrupt is accepted is saved to SSR. H'5E0 is set to INTEV INTEVT2. The BL, MD, and RB bits in SR are set to 1 and a branch occurs to V H'0600. See section 6, Interrupt Controller (INTC), for more information.

4.6 Cautions

- Return from exception handling
  - Check the BL bit in SR with software. When SPC and SSR have been saved to e memory, set the BL bit in SR to 1 before restoring them.
  - Issue an RTE instruction, which sets SPC in PC and SSR in SR, and causes a bra SPC address, and return from exception handling.

- Operation when exception or interrupt occurs while SR.BL = 1
  - Interrupt: Acceptance is suppressed until the BL bit in SR is cleared to 0. If there interrupt request and the reception conditions are satisfied, the interrupt is accep the execution of the instruction that clears the BL bit in SR to 0. In sleep or stan however, the interrupt will be accepted even when the BL bit in SR is 1.
  - Exception: No user break point trap will occur even when the break conditions a When one of the other exceptions occurs, a branch is made to the fixed address o (H'A0000000). In this case, the values of the EXPEVT, SPC, and SSR registers undefined. Differently from general reset processing, the $\overline{\text{RESETOUT}}$ pin is not and reset status is output from the STATUS0 and STATUS1 pins.

instruction, however, the branch destination PC is set in SPC. If the condition c
conditional delayed branch instruction is not satisfied, the delay slot PC is set i

- Initial register values after reset
  - — Undefined registers
    R0_BANK0/1–R7_BANK0/1, R8–R15, GBR, SPC, SSR, MACH, MACL, PR
  - — Initialized registers
    VBR = H'00000000
    SR.MD = 1, SR.BL = 1, SR.RB = 1, SR.I3–SR.I0 = H'F. Other SR bits are un
    PC = H'A0000000

- Ensure that an exception is not generated at an RTE instruction delay slot, as opera
guaranteed in this case.

- When the BL bit in the SR register is set to 1, ensure that a TLB-related exception
error does not occur at an LDC instruction that updates the SR register and the foll
instruction. This will be identified as the occurrence of multiple exceptions, and m
reset processing.

Table 5.1 Cache Specifications

Parameter	Specification
Capacity	16 kbytes
Structure	Instruction/data mixed, 4-way set associative
Locking	Way 2 and way 3 are lockable
Line size	16 bytes
Number of entries	256 entries/way
Write system	P0, P1, P3, U0: Write-back/write-through selectable
Replacement method	Least-recently-used (LRU) algorithm

5.1.2 Cache Structure

The cache mixes data and instructions and uses a 4-way set associative system. It is co
four ways (banks), each of which is divided into an address section and a data section.
address and data sections is divided into 256 entries. The data section of the entry is ca
Each line consists of 16 bytes (4 bytes × 4). The data capacity per way is 4 kbytes (16
entries), with a total of 16 kbytes in the cache as a whole (4 ways). Figure 5.1 shows th
structure.

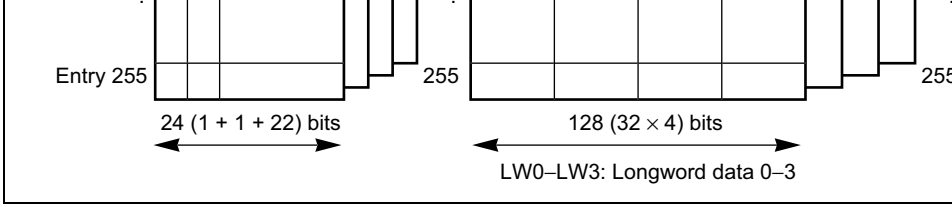


Figure 5.1 Cache Structure

Address Array: The V bit indicates whether the entry data is valid. When the V bit is 1 valid; when 0, data is not valid. The U bit indicates whether the entry has been written t back mode. When the U bit is 1, the entry has been written to; when 0, it has not. The a holds the physical address used in the external memory access. It is composed of 22 bit bits 31–10) used for comparison during cache searches.

In the SH7709S, the top three of 32 physical address bits are used as shadow bits (see s Bus State Controller (BSC)), and therefore in a normal replace operation the top three b tag address are cleared to 0.

The V and U bits are initialized to 0 by a power-on reset, but are not initialized by a ma The tag address is not initialized by either a power-on or manual reset.

Data Array: Holds a 16-byte instruction or data. Entries are registered in the cache in l (16 bytes). The data array is not initialized by a power-on or manual reset.

LRU: With the 4-way set associative system, up to four instructions or data with the sa address (address bits 11–4) can be registered in the cache. When an entry is registered, shows which of the four ways it is recorded in. There are six LRU bits, controlled by ha least-recently-used (LRU) algorithm is used to select the way.

The way that is to be replaced on a cache miss is determined by the 6-bit LRU. Table 5 the correspondence between the LRU bits and the way to be replaced when the cache-lo function is not used (when the cache-lock function is used, refer to section 5.2.2, Cache Register 2 (CCR2)). If a bit pattern other than those listed in table 5.2 is set in the LRU software, the cache will not function correctly. When modifying the LRU bits by softw of the patterns listed in table 5.2.

111000, 111001, 111011, 111100, 111110, 111111	0

5.1.3 Register Configuration

Table 5.3 shows details of the cache control register.

Table 5.3 Register Configuration

Register	Abbr.	R/W	Initial Value	Address	A
Cache control register	CCR	R/W	H'00000000	H'FFFFFFEC	3
Cache control register 2	CCR2	R/W	H'00000000	H'040000B0 (H'A40000B0)*	3

Note: * When address translation by the MMU does not apply, the address in parenth be used.

5.2 Register Description

5.2.1 Cache Control Register (CCR)

The cache is enabled or disabled using the CE bit of the cache control register (CCR). has a CF bit (which invalidates all cache entries), and a WT and CB bits (which select through mode or write-back mode). Programs that change the contents of the CCR reg be placed in address space that is not cached. When updating the contents of the CCR always set bits 4 to 0. Figure 5.2 shows the configuration of the CCR register.

WT: Write-through bit. Indicates the cache's operating mode for area P0, U0, and P3.
1 = write-through mode, 0 = write-back mode.

CE: Cache enable bit. Indicates whether the cache function is used.
1 = cache used, 0 = cache not used.

Figure 5.2 CCR Register Configuration

5.2.2 Cache Control Register 2 (CCR2)

CCR2 is used to control the cache-lock function and is valid only in cache locking mod locking mode means that the cache lock bit (bit 12) in SR (status register) is set to 1. Th lock function is invalid in non-cache locking mode (the cache-lock bit is 0).

When a prefetch instruction (PREF) is executed in cache locking mode and a cache mis one line size of data pointed to by Rn is brought to cache according to the setting of bit (W3LOAD and W3LOCK) and bits 1 and 0 (W2LOAD and W2LOCK) in CCR2. Tabl shows the relationship between the bit setting and way to be replaced when a prefetch i is executed. When a prefetch instruction is executed and there is a cache hit, new data is fetched and an entry which has already been valid is retained. For example, when the ca W3LOAD, and W3LOCK bits are set to 1 and a prefetch instruction is executed while size of data pointed to by Rn is already in way 0, a cache hit occurs and data is not fetch 3.

When cache is accessed by means of instructions except for a prefetch instruction in ca mode, a way that is replaced by the W3LOCK and W2LOCK bits is restricted. Table 5. the relationship between the bit setting of CCR2 and way to be replaced.

The program which modifies the contents of CCR2 must be placed in an address space not cache.

Figure 5.3 shows the configuration of CCR2.

CCR2 is a write-only register; if read, an undefined value will be returned.

W3LOCK: Way 3 lock bit. W3LOAD: Way 3 load bit.

When W3LOCK = 1 & W3LOAD = 1 & SR, CL = 1, the prefetched data will always be loaded into Way3. In all other conditions the prefetched data will be loaded into the way pointed by LRU.

Note: W2LOAD and W3LOAD should not be set to high at the same time.

—: Reserved bits.

Figure 5.3 CCR2 Register Configuration

Whenever CCR2 bit 8 (W3LOCK) or bit 0 (W2LOCK) is high the cache is locked. T data will not be overwritten unless W3LOCK bit and W2LOCK bit are reset or the PR condition during DSP mode matched. During cache locking mode, the LRU in table 5 replaced by tables 5.4 to 5.8.

Table 5.4 Way Replacement when PREF Instruction Ended Up in a Cache Mi

DSP bit	W3LOAD	W3LOCK	W2LOAD	W2LOCK	Way to be replaced
0	*	*	*	*	Depends on LRU (tabl
1	*	0	*	0	Depends on LRU (tabl
1	*	0	0	1	Depends on LRU (tabl
1	0	1	*	0	Depends on LRU (tabl
1	0	1	0	1	Depends on LRU (tabl
1	0	*	1	1	Way 2
1	1	1	0	*	Way 3

*: Don't care

Do not set as W3LOAD=1 and also W2LOAD=1

1	1	1	Depends on LRU (table

*: Don't care

Do not set as W3LOAD=1 and also W2LOAD=1

Table 5.6 LRU and Way Replacement (when W2LOCK=1)

LRU (5–0)	Way to be R
000000, 000001, 000100, 010100, 100000, 100001, 110000, 110100	3
000011, 000110, 000111, 001011, 001111, 010110, 011110, 011111	1
101001, 101011, 111000, 111001, 111011, 111100, 111110, 111111	0

Table 5.7 LRU and Way Replacement (when W3LOCK=1)

LRU (5–0)	Way to be R
000000, 000001, 000011, 001011, 100000, 100001, 101001, 101011	2
000100, 000110, 000111, 001111, 010100, 010110, 011110, 011111	1
110000, 110100, 111000, 111001, 111011, 111100, 111110, 111111	0

Table 5.8 LRU and Way Replacement (when W2LOCK=1 and W3LOCK=1)

LRU (5–0)	Way to be R
000000, 000001, 000011, 000100, 000110, 000111, 001011, 001111, 010100, 010110, 011110, 011111	1
100000, 100001, 101001, 101011, 110000, 110100, 111000, 111001, 111011, 111100, 111110, 111111	0

Entries are selected using bits 11–4 of the address (virtual) of the access to memory ar
address tag of that entry is read. In parallel to reading of the address tag, the virtual ad
translated to a physical address in the MMU. The physical address after translation an
physical address read from the address section are compared. The address comparison
ways. When the comparison shows a match and the selected entry is valid (V = 1), a c
occurs. When the comparison does not show a match or the selected entry is not valid
cache miss occurs. Figure 5.4 shows a hit on way 1.

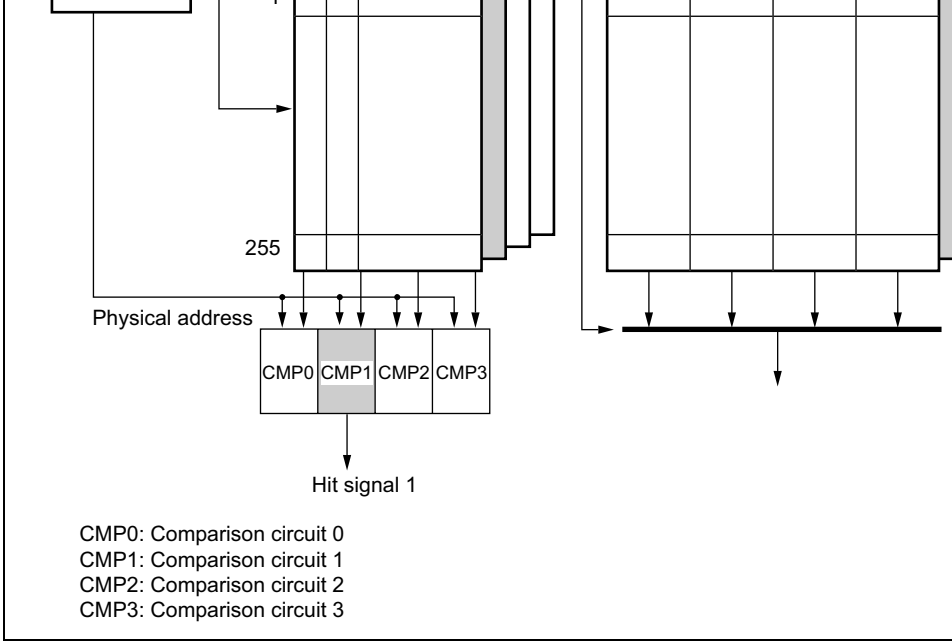


Figure 5.4 Cache Search Scheme (Normal Mode)

cleared to 0 and the V bit is set to 1.

5.3.3 Prefetch Operation

Prefetch Hit: The LRU will be updated to correctly indicate the latest way to have be contents of the cache will remain unchanged. Neither instructions nor data are transfer CPU.

Prefetch Miss: Neither instructions nor data are transferred to the CPU, and way repla takes place as shown in table 5.4. All other action is the same as for a read miss.

5.3.4 Write Access

Write Hit: In a write access in the write-back mode, the data is written to the cache ar of the entry written is set to 1. Writing occurs only to the cache; no external memory v issued. In the write-through mode, the data is written to the cache and an external mem cycle is issued.

Write Miss: In the write-back mode, an external write cycle starts when a write miss the entry is updated. The way to be replaced is shown in table 5.5. When the U bit of t be replaced is 1, the cache fill cycle starts after the entry is transferred to the write-bac The write-back unit is 16 bytes. Data is written to the cache and the U bit is set to 1. A cache completes its fill cycle, the write-back buffer writes back the entry to the memo write-through mode, no write to cache occurs in a write miss; the write is only to the e memory.

5.3.5 Write-Back Buffer

When the U bit of the entry to be replaced in the write-back mode is 1, it must be writt the external memory. To increase performance, the entry to be replaced is first transfe write-back buffer and fetching of new entries to the cache takes priority over writing b external memory. During the write back cycles, the cache can be accessed. The write- can hold one line of the cache data (16 bytes) and its physical address. Figure 5.5 shov configuration of the write-back buffer.

5.3.6 Coherency of Cache and External Memory

Use software to ensure coherency between the cache and the external memory. When m
shared by this LSI and another device is placed in an address space to be cached, the m
allocation cache is manipulated if necessary and a write back should be performed by i
the entry. This is also applied to memory shared by the CPU and DMAC in this LSI.

5.4 Memory-Mapped Cache

To allow software management of the cache, cache contents can be read and written by
MOV instructions in the privileged mode. The cache is mapped onto the P4 area in virt
space. The address array is mapped onto addresses H'F0000000 to H'F0FFFFFF, and th
array onto addresses H'F1000000 to H'F1FFFFFF. Only longword can be used as the ac
for the address array and data array, and instruction fetches cannot be performed.

5.4.1 Address Array

The address array is mapped to H'F0000000 to H'F0FFFFFF. The 32-bit address field (
read/write accessed) and 32-bit data field (for write access) must be specified to access
of the address array. The address field specifies information that selects the entry to be
the data field specifies the tag address, V bit, U bit, and LRU bits to be written to the ac
(figure 5.6 (1)).

In the address field, specify the entry's address in bits 11-4 to select the entry, W in bits
select the way, the A bit (bit 3) to specify an associative operation, and H'F0 in bits 31-
indicate access to the address array. Settings for the W bits (13-12) are as follows: 00 is
is way 1, 10 is way 2, and 11 is way 3.

In the data field, specify the tag address in bits 31-10, LRU in bits 9-4, U bit in bit 1, an
bit 0. The upper 3 bits (bit 31-29) of the tag address must always be 0.

corresponds to the entry address and way that were specified in the address field. The
bit (A bit) of the address field must be set to 0. An attempt to write to a cache line for
the U bit and V bit are set results in a write-back for that cache line. The tag address, I
and V bit specified in the data field are then written. Note that, when a 0 is written to t
should always be written to the U bit of the same entry, too.

(3) Address Array Write (with Associative Operation)

The associative bit (A bit) in the address field indicates whether the addresses are com
during writing. With the A bit set to 1, all 4 ways for the entry specified in the address
compared to the tag address specified in the data field for a match. The values of the U
bit specified in the data field will be written to the way that has a hit. However, the tag
the LRU will not be changed. If no way receives a hit, writing does not take place and
no operation.

This operation is used to invalidate the address specification for a cache. Write back w
place when the U bit of the entry that received a hit is 1. Note that, when a 0 is written
a 0 should always be written to the U bit of the same entry, too.

5.4.2 Data Array

The address array is mapped to H'F1000000 to H'F1FFFFFF. To access an element of
array, the 32-bit address field (for read/write access) and 32-bit data field (for write ac
be specified. The address field specifies the information that selects the entry to be acc
data field specifies the longword data to be written to the data array.

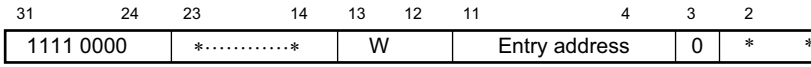
In the address field, specify the entry's address in bits 11-4, L in bits 3-2 to indicate th
position within a line (which consists of 16 bytes), W in bits 13-12 to select the way, a
bits 31-24 to indicate access to the data array. The L bits (3-2) specification is in the fo
form: 00 is longword 0, 01 is longword 1, 10 is longword 2, and 11 is longword 3. Set
W bits (13-12) are as follows: 00 is way 0, 01 is way 1, 10 is way 2, and 11 is way 3. S
is not allowed crossing longword boundaries, always set 00 in bits 1-0 of the address f

and way that were specified in the address field. The longword data will be written to th
the position selected by the L bits (3-2) of the address field.

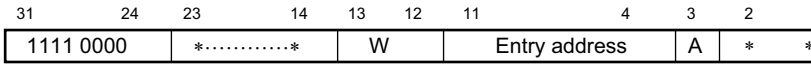
1. Address array access

Address specification

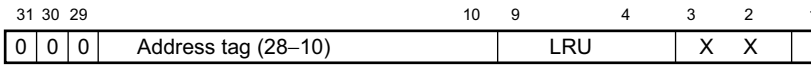
Read access



Write access

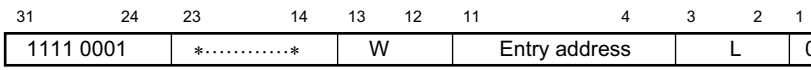


Data specification (both read and write accesses)

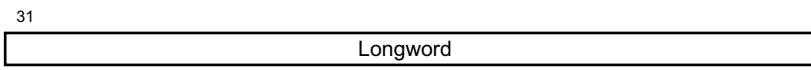


2. Data array access (both read and write accesses)

Address specification



Data specification



X: 0 for read, don't care for write
*: Don't care bit

Figure 5.6 Specifying Address and Data for Memory-Mapped Cache Acc

```
; R0 = H'0000 0000 LRU = H'000, U = 0, V = 0
; R1 = H'F000 1080, Way = 1, Entry = H'08, A = 0
;
    MOV.L  R0, @R1
```

To invalidate all entries and ways, write 0 to the following addresses.

Addresses

```
F000 0000
F000 0010
F000 0020
    :
F000 3FF0
```

This involves a total of 1,024 writes.

The above operation should be performed using a non-cacheable area.

(2) Invalidating a Specific Address

A specific address can be invalidated by writing a 0 to the entry's U and V bits. When 1, the tag address specified by the write data is compared to the tag address within the selected by the entry address. If the tag addresses match, data is written to the memory address. If no match is found, no operation is carried out. If the entry's U bit is 1 at tha entry is written back.

```
; R0 = H'0110 0010; Tag address = B'0000 0001 0001 0000 0000 0
V = 0
; R1 = H'F000 0088; Address array access, Entry = H'08, A = 1
;
    MOV.L  R0, @R1
```

```
    AND   R0, R3          ; The tag address is fetched. U = V =
    MOV.L R3, @R2         ; Associative purge.
```

The above operation should be performed using a non-cacheable area.

(3) Reading Data from a Specific Entry

This example reads the data section of a specific entry. The longword in the data field o
array in figure 5.6 is read to the register.

```
; R0 = H'F100 004C; Data array access, Entry = H'04,
; Way = 0, Longword address = 3
;
   MOV.L  R0, @R1      ; Longword 3 is read.
```

6.1.1 Features

The INTC has the following features:

- 16 levels of interrupt priority can be set: By setting the five interrupt-priority regis priorities of on-chip peripheral module, IRQ, and PINT interrupts can be selected f levels for individual request sources.
- NMI noise canceler function: An NMI input-level bit indicates the NMI pin state. this bit in the interrupt exception service routine, the pin state can be checked, enal used as a noise canceler.
- External devices can be notified that an interrupt has been received ($\overline{\text{IRQOUT}}$): W SH7709S has released the bus, the external bus master can be notified that an exte interrupt, an on-chip peripheral module interrupt, or a memory refresh request has enabling the bus to be requested.

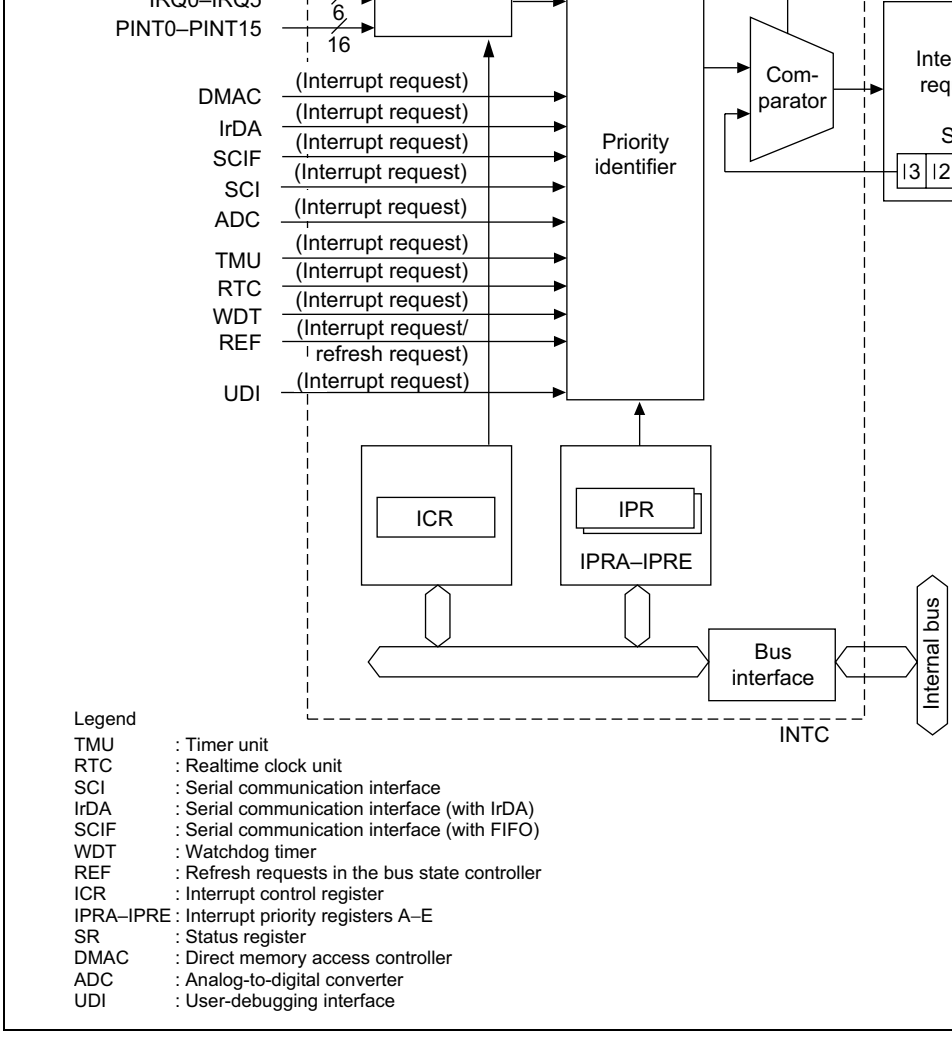


Figure 6.1 Block Diagram of INTC

RENESAS

			SR.
Interrupt input pins	IRQ5–IRQ0 $\overline{\text{IRL3}}$–$\overline{\text{IRL0}}$ $\overline{\text{IRLS3}}$-$\overline{\text{IRLS0}}$	I	Input of interrupt request s maskable by the interrupt SR.
Port interrupt input pins	PINT0–PINT15	I	Input of port interrupt requ maskable by the interrupt SR.
Bus request output pin	$\overline{\text{IRQOUT}}$	O	Output of signal that notifie devices that an interrupt s memory refresh has occur

Interrupt control register 1	ICR1	R/W	H'0000	H'04000010 (H'A4000010)*3
Interrupt control register 2	ICR2	R/W	H'0000	H'04000012 (H'A4000012)*3
PINT interrupt enable register	PINTER	R/W	H'0000	H'04000014 (H'A4000014)*3
Interrupt priority register A	IPRA	R/W	H'0000	H'FFFFFEE2
Interrupt priority register B	IPRB	R/W	H'0000	H'FFFFFEE4
Interrupt priority register C	IPRC	R/W	H'0000	H'04000016 (H'A4000016)*3
Interrupt priority register D	IPRD	R/W	H'0000	H'04000018 (H'A4000018)*3
Interrupt priority register E	IPRE	R/W	H'0000	H'0400001A (H'A400001A)*3
Interrupt request register 0	IRR0	R/W	H'00	H'04000004 (H'A4000004)*3
Interrupt request register 1	IRR1	R	H'00	H'04000006 (H'A4000006)*3
Interrupt request register 2	IRR2	R	H'00	H'04000008 (H'A4000008)*3

Notes: 1. Initialized by a power-on or manual reset.
2. H'8000 when the NMI pin is high, H'0000 when the NMI pin is low.
3. When address translation by the MMU does not apply, the address in parent should be used.

The NMI interrupt has the highest priority level of 16. When the BLMSK bit in the int
control register (ICR1) is 1 or the BL bit in the status register (SR) is 0, NMI interrupt
accepted when the MAI bit in the ICR1 register is 0. NMI interrupts are edge-detected
standby mode, the interrupt is accepted regardless of the BL setting. The NMI edge se
(NMIE) in the interrupt control register 0 (ICR0) is used to select either rising or fallin
detection. When the NMIE bit in the ICR0 register is changed, an NMI interrupt is not
20 cycles after changing ICR0. NMIE to avoid a false detection of NMI. NMI interrup
handling does not affect the interrupt mask level bits (I3–I0) in the status register (SR)

When the BL bit is land the BLMSK bit in the ICR1 register is set to 1 and only NMI
are accepted, the SPC register and SSR register are updated by the NMI interrupt hand
it impossible to return to the original processing from exception handling initiated prio
NMI interrupt. Use should therefore be restricted to cases where return is not necessar

It is possible to wake the chip up from the standby state with an NMI interrupt (except
MAI bit in the ICR1 register is set to 1).

6.2.2 IRQ Interrupts

IRQ interrupts are input by level or edge from pins IRQ0–IRQ5. The priority level can
interrupt priority registers C–D (IPRC–IPRD) in a range from 0 to 15.

When using edge-sensing for IRQ interrupts, clear the interrupt source by having softw
from the corresponding bit in IRR0, then write 0 to the bit.

When the ICR1 register is rewritten, IRQ interrupts may be mistakenly detected, depe
pin states. To prevent this, rewrite the register while interrupts are masked, then releas
after clearing the illegal interrupt by writing 0 to interrupt request register 0 (IRR0).

Edge input interrupt detection requires input of a pulse width of more than two cycles
peripheral clock (Pϕ) basis.

The interrupt mask bits (I3–I0) in the status register (SR) are not affected by IRQ inter
handling.

processing (WDT count).

6.2.3 IRL Interrupts

IRL interrupts are input by level at pins $\overline{IRL3}$–$\overline{IRL0}$ and $\overline{IRLS3}$–$\overline{IRLS0}$. $\overline{IRLS3}$–$\overline{IRLS0}$
enabled when the IRQLVL bit and IRLSEN bit in interrupt control register 1 (ICR1) ar
The priority level is the higher level indicated by pins $\overline{IRL3}$–$\overline{IRL0}$ and $\overline{IRLS3}$–$\overline{IRLS0}$. A
$\overline{IRL0}$/$\overline{IRLS3}$–$\overline{IRLS0}$ value of 0 (0000) indicates the highest-level interrupt request (inte
priority level 15). A value of 15 (1111) indicates no interrupt request (interrupt priority
Figure 6.2 shows an example of IRL interrupt connection. Table 6.3 shows $\overline{IRL}$/$\overline{IRLS}$ p
interrupt levels.

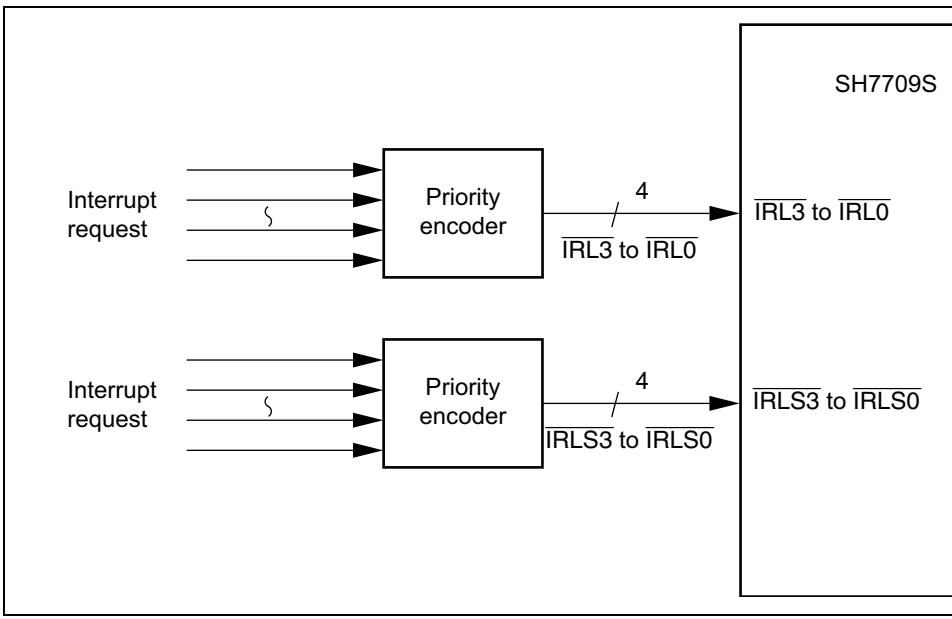


Figure 6.2 Example of IRL Interrupt Connection

0	1	0	1	10	Level 10 interrup
0	1	1	0	9	Level 9 interrupt
0	1	1	1	8	Level 8 interrupt
1	0	0	0	7	Level 7 interrupt
1	0	0	1	6	Level 6 interrupt
1	0	1	0	5	Level 5 interrupt
1	0	1	1	4	Level 4 interrupt
1	1	0	0	3	Level 3 interrupt
1	1	0	1	2	Level 2 interrupt
1	1	1	0	1	Level 1 interrupt
1	1	1	1	0	No interrupt requ

A noise-cancellation feature is built in, and the IRL interrupt is not detected unless the sampled at every peripheral module clock cycle remain unchanged for two consecutiv that no transient level on the $\overline{\text{IRL}}$/$\overline{\text{IRLS}}$ pin change is detected. In standby mode, as the clock is stopped, noise cancellation is performed using the 32-kHz clock for the RTC. Therefore when the RTC is not used, interruption by means of IRL interrupts cannot b in standby mode.

The priority level of the IRL interrupt must not be lowered until the interrupt is accept interrupt handling starts. Correct operation cannot be guaranteed if the level is not mai However, the priority level can be changed to a higher one.

The interrupt mask bits (I3–I0) in the status register (SR) are not affected by $\overline{\text{IRL}}$/$\overline{\text{IRL}}$ handling.

When the interrupt level of the IRL interrupt is higher than the level set by the I3-I0 b the IRL interrupt can be used to recover from standby mode (however, this only applie RTC is used for 32-kHz oscillator).

The interrupt mask bits (I3–I0) in the status register (SR) are not affected by PINT interr
handling.

PINT0/1 interrupts can wake the chip up from the standby state when the relevant interr
higher than the setting of I3–I0 in the SR register (but only when the RTC 32-kHz oscil
used).

6.2.5 On-Chip Peripheral Module Interrupts

On-chip peripheral module interrupts are generated by the following ten modules:

- Timer unit (TMU)
- Realtime clock (RTC)
- Serial communication interfaces (SCI, IrDA, SCIF)
- Bus state controller (BSC)
- Watchdog timer (WDT)
- Direct memory access controller (DMAC)
- Analog-to-digital converter (ADC)
- User-debugging interface (UDI)

Not every interrupt source is assigned a different interrupt vector. Sources are reflected
interrupt event registers (INTEVT and INTEVT2). It is easy to identify sources by usin
of the INTEVT or INTEVT2 register as a branch offset.

A priority level (from 0 to 15) can be set for each module except UDI by writing to inte
priority registers A, B, and E (IPRA, IPRB, and IPRE). The priority level of the UDI in
15 (fixed).

The interrupt mask bits (I3–I0) in the status register are not affected by on-chip periphe
interrupt handling.

TMU and RTC interrupts can wake the chip up from the standby state when the relevan
level is higher than the setting of I3–I0 in the SR register (but only when the RTC 32-k
oscillator is used).

levels 0–15 as required by using interrupt priority registers A–E (IPRA–IPRE). The p
on-chip peripheral module, IRQ, and PINT interrupts is set to 0 by a reset.

When the priorities of multiple interrupt sources are set to the same level and such inte
generated simultaneously, they are handled according to the default order shown in tab
6.5.

	IRQ2	H'200–3C0* (H'640)	0–15 (0)	IPRC (11–8)	—
	IRQ3	H'200–3C0* (H'660)	0–15 (0)	IPRC (15–12)	—
	IRQ4	H'200–3C0* (H'680)	0–15 (0)	IPRD (3–0)	—
	IRQ5	H'200–3C0* (H'6A0)	0–15 (0)	IPRD (7–4)	—
PINT	PINT0–7	H'200–3C0* (H'700)	0–15 (0)	IPRD (15–12)	—
	PINT8–15	H'200–3C0* (H'720)	0–15 (0)	IPRD (11–8)	—
DMAC	DEI0	H'200–3C0* (H'800)	0–15 (0)	IPRE (15–12)	High
	DEI1	H'200–3C0* (H'820)			
	DEI2	H'200–3C0* (H'840)			
	DEI3	H'200–3C0* (H'860)			Low
IrDA	ERI1	H'200–3C0* (H'880)	0–15 (0)	IPRE (11–8)	High
	RXI1	H'200–3C0* (H'8A0)			
	BRI1	H'200–3C0* (H'8C0)			
	TXI1	H'200–3C0* (H'8E0)			Low
SCIF	ERI2	H'200–3C0* (H'900)	0–15 (0)	IPRE (7–4)	High
	RXI2	H'200–3C0* (H'920)			
	BRI2	H'200–3C0* (H'940)			
	TXI2	H'200–3C0* (H'960)			Low
ADC	ADI	H'200–3C0* (H'980)	0–15 (0)	IPRE (3–0)	—
TMU0	TUNI0	H'400 (H'400)	0–15 (0)	IPRA (15–12)	—
TMU1	TUNI1	H'420 (H'420)	0–15 (0)	IPRA (11–8)	—
TMU2	TUNI2	H'440 (H'440)	0–15 (0)	IPRA (7–4)	High
	TICPI2	H'460 (H'460)			Low

	TXI	H'520 (H'520)			↓
	TEI	H'540 (H'540)			Low
WDT	ITI	H'560 (H'560)	0–15 (0)	IPRB (15–12)	—
REF	RCMI	H'580 (H'580)	0–15 (0)	IPRB (11–8)	High
	ROVI	H'5A0 (H'5A0)			Low

Note: * The code corresponding to an interrupt level shown in table 6.6 is set.

	$\overline{IRL(3:0)}^{*2}$ = 0010	H'240 (H'240)	13	—	—
	$\overline{IRL(3:0)}^{*2}$ = 0011	H'260 (H'260)	12	—	—
	$\overline{IRL(3:0)}^{*2}$ = 0100	H'280 (H'280)	11	—	—
	$\overline{IRL(3:0)}^{*2}$ = 0101	H'2A0 (H'2A0)	10	—	—
	$\overline{IRL(3:0)}^{*2}$ = 0110	H'2C0 (H'2C0)	9	—	—
	$\overline{IRL(3:0)}^{*2}$ = 0111	H'2E0 (H'2E0)	8	—	—
	$\overline{IRL(3:0)}^{*2}$ = 1000	H'300 (H'300)	7	—	—
	$\overline{IRL(3:0)}^{*2}$ = 1001	H'320 (H'320)	6	—	—
	$\overline{IRL(3:0)}^{*2}$ = 1010	H'340 (H'340)	5	—	—
	$\overline{IRL(3:0)}^{*2}$ = 1011	H'360 (H'360)	4	—	—
	$\overline{IRL(3:0)}^{*2}$ = 1100	H'380 (H'380)	3	—	—
	$\overline{IRL(3:0)}^{*2}$ = 1101	H'3A0 (H'3A0)	2	—	—
	$\overline{IRL(3:0)}^{*2}$ = 1110	H'3C0 (H'3C0)	1	—	—
IRQ	IRQ4	H'200–3C0[*1] (H'680)	0–15 (0)	IPRD (3–0)	—
	IRQ5	H'200–3C0[*1] (H'6A0)	0–15 (0)	IPRD (7–4)	—
PINT	PINT0–7	H'200–3C0[*1] (H'700)	0–15 (0)	IPRD (15–12)	—
	PINT8–15	H'200–3C0[*1] (H'720)	0–15 (0)	IPRD (11–8)	—
DMAC	DEI0	H'200–3C0[*1] (H'800)	0–15 (0)	IPRE (15–12)	High
	DEI1	H'200–3C0[*1] (H'820)			↕
	DEI2	H'200–3C0[*1] (H'840)			
	DEI3	H'200–3C0[*1] (H'860)			Low
IrDA	ERI1	H'200–3C0[*1] (H'880)	0–15 (0)	IPRE (11–8)	High
	RXI1	H'200–3C0[*1] (H'8A0)			↕
	BRI1	H'200–3C0[*1] (H'8C0)			
	TXI1	H'200–3C0[*1] (H'8E0)			Low

TMU0	TUNI0	H'400 (H'400)	0–15 (0)	IPRA (15–12)	—
TMU1	TUNI1	H'420 (H'420)	0–15 (0)	IPRA (11–8)	—
TMU2	TUNI2	H'440 (H'440)	0–15 (0)	IPRA (7–4)	High
	TICPI2	H'460 (H'460)			Low
RTC	ATI	H'480 (H'480)	0–15 (0)	IPRA (3–0)	High
	PRI	H'4A0 (H'4A0)			
	CUI	H'4C0 (H'4C0)			Low
SCI	ERI	H'4E0 (H'4E0)	0–15 (0)	IPRB (7–4)	High
	RXI	H'500 (H'500)			
	TXI	H'520 (H'520)			
	TEI	H'540 (H'540)			Low
WDT	ITI	H'560 (H'560)	0–15 (0)	IPRB (15–12)	—
REF	RCMI	H'580 (H'580)	0–15 (0)	IPRB (11–8)	High
	ROVI	H'5A0 (H'5A0)			Low

Notes: 1. The code corresponding to an interrupt level shown in table 6.6 is set.
2. When $\overline{\text{IRLS3}}$–$\overline{\text{IRLS0}}$ are enabled, IRL is the higher level of $\overline{\text{IRL3}}$–$\overline{\text{IRL0}}$ and $\overline{\text{IRLS0}}$.

10	H'2A0
9	H'2C0
8	H'2E0
7	H'300
6	H'320
5	H'340
4	H'360
3	H'380
2	H'3A0
1	H'3C0

Bit:	15	14	13	12	11	10	9
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 6.7 lists the relationship between the interrupt sources and the IPRA—IPRE bits

Table 6.7 Interrupt Request Sources and IPRA–IPRE

Register	Bits 15 to 12	Bits 11 to 8	Bits 7 to 4	Bits 3
IPRA	TMU0	TMU1	TMU2	RTC
IPRB	WDT	REF	SCI0	Reserv
IPRC	IRQ3	IRQ2	IRQ1	IRQ0
IPRD	PINT0 to PINT7	PINT8 to PINT15	IRQ5	IRQ4
IPRE	DMAC	IrDA	SCIF	ADC

Note: * Always read as 0. Only 0 should be written.

As shown in table 6.7, on-chip peripheral module, IRQ, or PINT interrupts are assigne bit groups in each register. These 4-bit groups (bits 15 to 12, bits 11 to 8, bits 7 to 4, a 0) are set with values from H'0 (0000) to H'F (1111). Setting H'0 means priority level is requested); H'F is priority level 15 (the highest level). A reset initializes IPRA–IPR

R/W:	R	R	R	R	R	R	R
Bit:	7	6	5	4	3	2	1
	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R

Note: * 1 when NMI input is high, 0 when NMI input is low.

Bit 15—NMI Input Level (NMIL): Sets the level of the signal input at the NMI pin. T be read to determine the NMI pin level. This bit cannot be modified.

Bit 15: NMIL	Description
0	NMI input level is low
1	NMI input level is high

Bit 8—NMI Edge Select (NMIE): Selects whether the falling or rising edge of the int request signal at the NMI pin is detected.

Bit 8: NMIE	Description
0	Interrupt request is detected on falling edge of NMI input
1	Interrupt request is detected on rising edge of NMI input

Bits 14 to 9 and 7 to 0—Reserved: These bits are always read as 0. The write value sh always be 0.

R/W:	R/W	R/W	R/W	RW	R/W	R/W	R/W
Bit:	7	6	5	4	3	2	1
	IRQ31S	IRQ30S	IRQ21S	IRQ20S	IRQ11S	IRQ10S	IRQ0
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 15—Mask All Interrupts (MAI): When set to 1, all interrupt requests are masked low level is being input to the NMI pin. Masks NMI interrupts in standby mode.

Bit 15: MAI	Description
0	All interrupt requests are not masked when NMI pin is low level
1	All interrupt requests are masked when NMI pin is low level

Bit 14—Interrupt Request Level Detect (IRQLVL): Selects whether the IRQ3–IRQ used as four independent interrupt pins or as 15-level interrupt pins encoded as $\overline{\text{IRL3}}$–

Bit 14: IRQLVL	Description
0	Used as four independent interrupt request pins IRQ3–IRQ0
1	Used as 15-level interrupt pins encoded as $\overline{\text{IRL3}}$–$\overline{\text{IRL0}}$

Bit 13—BL Bit Mask (BLMSK): Specifies whether NMI interrupts are masked when in the SR register is 1.

Bit 13: BLMSK	Description
0	NMI interrupts are masked when BL bit is 1
1	NMI interrupts are accepted regardless of BL bit setting

the IRQ5 pin is detected at the rising edge, at the falling edge, or at the low level.

Bit 11: IRQ51S	Bit 10: IRQ50S	Description
0	0	An interrupt request is detected at IRQ5 input falling (Ir
	1	An interrupt request is detected at IRQ5 input rising
1	0	An interrupt request is detected at IRQ5 input low lev
	1	Reserved

Bits 9 and 8—IRQ4 Sense Select (IRQ41S, IRQ40S): Select whether the interrupt sig
IRQ4 pin is detected at the rising edge, at the falling edge, or at the low level.

Bit 9: IRQ41S	Bit 8: IRQ40S	Description
0	0	An interrupt request is detected at IRQ4 input falling (Ir
	1	An interrupt request is detected at IRQ4 input rising
1	0	An interrupt request is detected at IRQ4 input low lev
	1	Reserved

Bits 7 and 6—IRQ3 Sense Select (IRQ31S, IRQ30S): Select whether the interrupt sig
IRQ3 pin is detected at the rising edge, at the falling edge, or at the low level.

Bit 7: IRQ31S	Bit 6: IRQ30S	Description
0	0	An interrupt request is detected at IRQ3 input falling (Ir
	1	An interrupt request is detected at IRQ3 input rising
1	0	An interrupt request is detected at IRQ3 input low lev
	1	Reserved

Bits 3 and 2—IRQ1 Sense Select (IRQ11S, IRQ10S): Select whether the interrupt s
IRQ1 pin is detected at the rising edge, at the falling edge, or at the low level.

Bit 3: IRQ11S	Bit 2: IRQ10S	Description
0	0	An interrupt request is detected at IRQ1 input fallin
	1	An interrupt request is detected at IRQ1 input risin
1	0	An interrupt request is detected at IRQ1 input low l
	1	Reserved

Bits 1 and 0—IRQ0 Sense Select (IRQ01S, IRQ00S): Select whether the interrupt s
IRQ0 pin is detected at the rising edge, at the falling edge, or at the low level.

Bit 1: IRQ01S	Bit 0: IRQ00S	Description
0	0	An interrupt request is detected at IRQ0 input fallin
	1	An interrupt request is detected at IRQ0 input risin
1	0	An interrupt request is detected at IRQ0 input low l
	1	Reserved

R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1
	PINT7S	PINT6S	PINT5S	PINT4S	PINT3S	PINT2S	PINT1S
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bits 15 to 0—PINT15 to PINT0 Sense Select (PINT15S to PINT0S): Select whether request signals to PINT15 to PINT0 are detected at the low level or high level.

Bits 15–0:

PINT15S to PINT0S	Description
0	Interrupt requests are detected at low level input to the PINT p (I
1	Interrupt requests are detected at high level input to the PINT p

R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit:	7	6	5	4	3	2	1
	PINT7E	PINT6E	PINT5E	PINT4E	PINT3E	PINT2E	PINT
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bits 15 to 0—PINT15 to PINT0 Interrupt Enable (PINT15E to PINT0E): Enable interrupt request input to pins PINT15 to PINT0.

Bits 15–0: PINT15E to PINT0E	**Description**
0	PINT input interrupt requests disabled
1	PINT input interrupt requests enabled

When all or some of pins PINT0–PINT15 are not used for interrupt input, bits corresp pins not used as interrupt request pins should be cleared to 0.

R/W:	R	R	R/W	R/W	R/W	R/W	R/W

When clearing an IRQ5R–IRQ0R bit to 0, read the bit while bit set to 1, and then write case, 0 should be written only to the bits to be cleared and 1 to the other bits. The conte bits to which 1 is written do not change.

Bit 7—PINT0 to PINT7 Interrupt Request (PINT0R): Indicates whether there is int request input to pins PINT0 to PINT7.

Bit 7: PINT0R	Description	
0	No interrupt request to pins PINT0 to PINT7	(Ir
1	Interrupt to pins PINT0 to PINT7	

Bit 6—PINT8 to PINT15 Interrupt Request (PINT1R): Indicates whether there is ir request input to pins PINT8 to PINT15.

Bit 6: PINT1R	Description	
0	No interrupt request input to pins PINT8 to PINT15	(Ir
1	Interrupt request input to pins PINT8 to PINT15	

Bit 5—IRQ5 Interrupt Request (IRQ5R): Indicates whether there is interrupt reques the IRQ5 pin. When edge detection mode is set for IRQ5, an interrupt request is cleared clearing the IRQ5R bit.

Bit 5: IRQ5R	Description	
0	No interrupt request input to IRQ5 pin	(Ir
1	Interrupt request input to IRQ5 pin	

Bit 3—IRQ3 Interrupt Request (IRQ3R): Indicates whether there is interrupt reque
the IRQ3 pin. When edge detection mode is set for IRQ3, an interrupt request is cleare
clearing the IRQ3R bit.

Bit 3: IRQ3R	Description	
0	No interrupt request input to IRQ3 pin	(
1	Interrupt request input to IRQ3 pin	

Bit 2—IRQ2 Interrupt Request (IRQ2R): Indicates whether there is interrupt reque
the IRQ2 pin. When edge detection mode is set for IRQ2, an interrupt request is cleare
clearing the IRQ2R bit.

Bit 2: IRQ2R	Description	
0	No interrupt request input to IRQ2 pin	(
1	Interrupt request input to IRQ2 pin	

Bit 1—IRQ1 Interrupt Request (IRQ1R): Indicates whether there is interrupt reque
the IRQ1 pin. When edge detection mode is set for IRQ1, an interrupt request is cleare
clearing the IRQ1R bit.

Bit 1: IRQ1R	Description	
0	No interrupt request input to IRQ1 pin	(
1	Interrupt request input to IRQ1 pin	

Bit 0—IRQ0 Interrupt Request (IRQ0R): Indicates whether there is interrupt reque
the IRQ0 pin. When edge detection mode is set for IRQ0, an interrupt request is cleare
clearing the IRQ0R bit.

Bit 0: IRQ0R	Description	
0	No interrupt request input to IRQ0 pin	(
1	Interrupt request input to IRQ0 pin	

R/W:	R	R	R	R	R	R	R

Bit 7—TXI1 Interrupt Request (TXI1R): Indicates whether a TXI1 (IrDA) interrupt been generated.

Bit 7: TXI1	Description
0	TXI1 interrupt request not generated (Ir
1	TXI1 interrupt request generated

Bit 6—BRI1 Interrupt Request (BRI1R): Indicates whether a BRI1 (IrDA) interrupt been generated.

Bit 6: BRI1R	Description
0	BRI1 interrupt request not generated (Ir
1	BRI1 interrupt request generated

Bit 5—RXI1 Interrupt Request (RXI1R): Indicates whether an RXI1 (IrDA) interrup has been generated.

Bit 5: RXI1R	Description
0	RXI1 interrupt request not generated (Ir
1	RXI1 interrupt request generated

Bit 4—ERI1 Interrupt Request (ERI1R): Indicates whether an ERI1 (IrDA) interrup has been generated.

Bit 4: ERI1R	Description
0	ERI1 interrupt request not generated (Ir
1	ERI1 interrupt request generated

has been generated.

Bit 2: DEI2R	Description
0	DEI2 interrupt request not generated
1	DEI2 interrupt request generated

Bit 1—DEI1 Interrupt Request (DEI1R): Indicates whether a DEI1 (DMAC) interr
has been generated.

Bit 1: DEI1R	Description
0	DEI1 interrupt request not generated
1	DEI1 interrupt request generated

Bit 0—DEI0 Interrupt Request (DEI0R): Indicates whether a DEI0 (DMAC) interr
has been generated.

Bit 0: DEI0R	Description
0	DEI0 interrupt request not generated
1	DEI0 interrupt request generated

6.3.8 Interrupt Request Register 2 (IRR2)

IRR2 is an 8-bit read-only register that indicates whether an A/D converter or SCIF in
request has been generated. This register is initialized to H'00 by a power-on reset or
but is not initialized in standby mode.

Bit:	7	6	5	4	3	2	1
	—	—	—	ADIR	TXI2R	BRI2R	RXI2
Initial value:	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R

Bit 3—TXI2 Interrupt Request (TXI2R): Indicates whether a TXI2 (SCIF) interrupt
been generated.

Bit 3: TXI2R	Description	
0	TXI2 interrupt request not generated	(In
1	TXI2 interrupt request generated	

Bit 2—BRI2 Interrupt Request (BRI2R): Indicates whether a BRI2 (SCIF) interrupt
been generated.

Bit 2: BRI2R	Description	
0	BRI2 interrupt request not generated	(In
1	BRI2 interrupt request generated	

Bit 1—RXI2 Interrupt Request (RXI2R): Indicates whether an RXI2 (SCIF) interrup
has been generated.

Bit 1: RXI2R	Description	
0	RXI2 interrupt request not generated	(In
1	RXI2 interrupt request generated	

Bit 0—ERI2 Interrupt Request (ERI2R): Indicates whether an ERI2 (SCIF) interrup
has been generated.

Bit 0: ERI2R	Description	
0	ERI2 interrupt request not generated	(In
1	ERI2 interrupt request generated	

following the priority levels set in interrupt priority registers A to E (IPRA to IPRI
priority interrupts are held pending. If two of these interrupts have the same priorit
multiple interrupts occur within a single module, the interrupt with the highest defa
or the highest priority within its IPR setting unit (as indicated in tables 6.4 and 6.5)

3. The priority level of the interrupt selected by the interrupt controller is compared v
interrupt mask bits (I3–I0) in the status register (SR) of the CPU. If the request pri
higher than the level in bits I3–I0, the interrupt controller accepts the interrupt and
interrupt request signal to the CPU. When the interrupt controller receives an inter
level is output from the $\overline{\text{IRQOUT}}$ pin.
4. Detection timing: The INTC operates, and notifies the CPU of interrupt requests, i
synchronization with the peripheral clock (Pϕ). The CPU receives an interrupt at a
instructions.
5. The interrupt source code is set in the interrupt event registers (INTEVT and INTE
6. The status register (SR) and program counter (PC) are saved to SSR and SPC, resp
7. The block bit (BL), mode bit (MD), and register bank bit (RB) in SR are set to 1.
8. The CPU jumps to the start address of the interrupt handler (the sum of the value s
vector base register (VBR) and H'00000600). This jump is not a delayed branch. T
handler may branch with the INTEVT and INTEVT2 register value as its offset in
identify the interrupt source. This enables it to branch to the handling routine for th
interrupt source.

Notes: 1. The interrupt mask bits (I3–I0) in the status register (SR) are not changed b
acceptance of an interrupt in the SH7709S.
2. $\overline{\text{IRQOUT}}$ outputs a low level until the interrupt request is cleared. Howeve
interrupt source is masked by an interrupt mask bit, the $\overline{\text{IRQOUT}}$ pin retur
high level. The level is output without regard to the BL bit.
3. The interrupt source flag should be cleared in the interrupt handler. To ens
interrupt request that should have been cleared is not inadvertently accepte
the interrupt source flag after it has been cleared, then wait for the interval
table 6.8 (Time for priority decision and SR mask bit comparison) before c
BL bit or executing an RTE instruction.

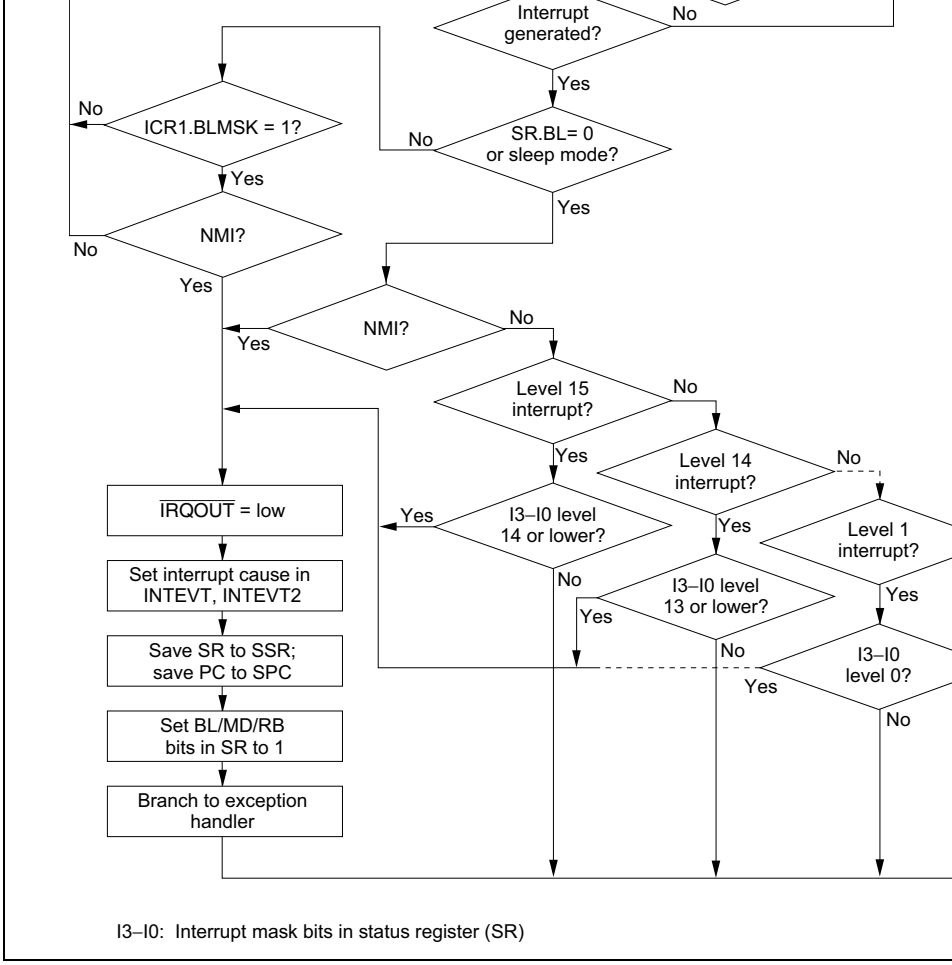


Figure 6.3 Interrupt Operation Flowchart

4. Clear the BL bit in SR, and set the accepted interrupt level in the interrupt mask bi
5. Handle the interrupt.
6. Execute the RTE instruction.

When these procedures are followed in order, an interrupt of higher priority than the o handled can be accepted after clearing BL in step 4. Figure 6.3 shows a sample interru flowchart.

6.5 Interrupt Response Time

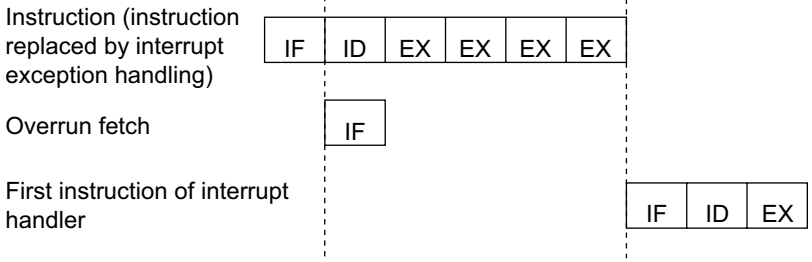
The time from generation of an interrupt request until interrupt exception handling is p and fetching of the first instruction of the exception handler is started (the interrupt res is shown in table 6.8. Figure 6.4 shows an example of pipeline operation when an IRL accepted. When SR.BL is 1, interrupt exception handling is masked, and is kept waitin completion of an instruction that clears BL to 0.

				+ 3 × Pcyc	
Wait time until end of sequence being executed by CPU	X (≥ 0) × Icyc	X (≥ 0) × Icyc	X (≥ 0) × Icyc	X (≥ 0) × Icyc	Interru handli waitin execu tion er numbe tion ex states maxim time is Howev set to ction e by an interru handli deferr compl instruc clears the fol instruc interru handli handli further
Time from interrupt exception handling (save of SR and PC) until fetch of first instruction of exception handler is started	5 × Icyc	5 × Icyc	5 × Icyc	5 × Icyc	

				+ 3 × Pcyc	
Minimum case*2	7.5	16.5	12.5	8.5*5/11.5*6	At 60 = 30 0.13-
Maximum case*3	8.5 + S	26.5 + S	18.5 + S	10.5 + S*5 16.5 + S*6	At 60 = 15 0.26- case cach At 60 = 15 0.29- (whe mem perfo wait

Icyc: Duration of one cycle of internal clock supplied to CPU.
Bcyc: Duration of one CKIO cycle.
Pcyc: Duration of one cycle of peripheral clock supplied to peripheral modules.

Notes: 1. S also includes the memory access wait time.
The processing requiring the maximum execution time is LDC.L @Rm+, SR
memory access is a cache-hit, this requires seven instruction execution cyc
the external access is performed, the corresponding number of cycles mus
There are also instructions that perform two external memory accesses; if t
memory access is slow, the number of instruction execution cycles will incr
accordingly.
2. The internal clock:CKIO:peripheral clock ratio is 2:1:1.
3. The internal clock:CKIO:peripheral clock ratio is 4:1:1.
4. IRQ mode
5. Modules: TMU, RTC, SCI, WDT, REFC
6. Modules: DMAC, ADC, IrDA, SCIF



IF: Instruction fetch: Instruction is fetched from memory in which program is sto
ID: Instruction decode: Fetched instruction is decoded.
EX: Instruction execution: Data operation and address calculation are performe

Figure 6.4 Example of Pipeline Operations when IRL Interrupt is Accep

instruction fetches.

7.1.1 Features

The user break controller has the following features:

- The following break comparison conditions can be set.

  Number of break channels: two channels (channels A and B)
  User break can be requested as either the independent or sequential condition on cl
  and B (sequential break setting: channel A and, then channel B match with logical
  not in the same bus cycle).

  Address (Compares 40 bits comprised of a 32-bit logical address prefixed with
  address. Comparison bits are maskable in 32-bit units, user can easily program
  addresses at bottom 12 bits (4-k page), bottom 10 bits (1-k page), or any size o

  One of two address buses (CPU address bus (LAB), cache address bus (IAB)) c
  selected.

  — Data (only on channel B, 32-bit maskable)
  — One of the two data buses (CPU data bus (LDB), cache data bus (IDB)) can be
  — Bus master: CPU cycle or DMAC cycle
  — Bus cycle: instruction fetch or data access
  — Read/write
  — Operand size: byte, word, or longword
- User break is generated upon satisfying break conditions. A user-designed user-bre
  condition exception processing routine can be run.
- In an instruction fetch cycle, it can be selected that a break is set before or after an
  is executed.
- Maximum repeat times for the break condition: $2^{12} - 1$ times.
- Eight pairs of branch source/destination buffers.

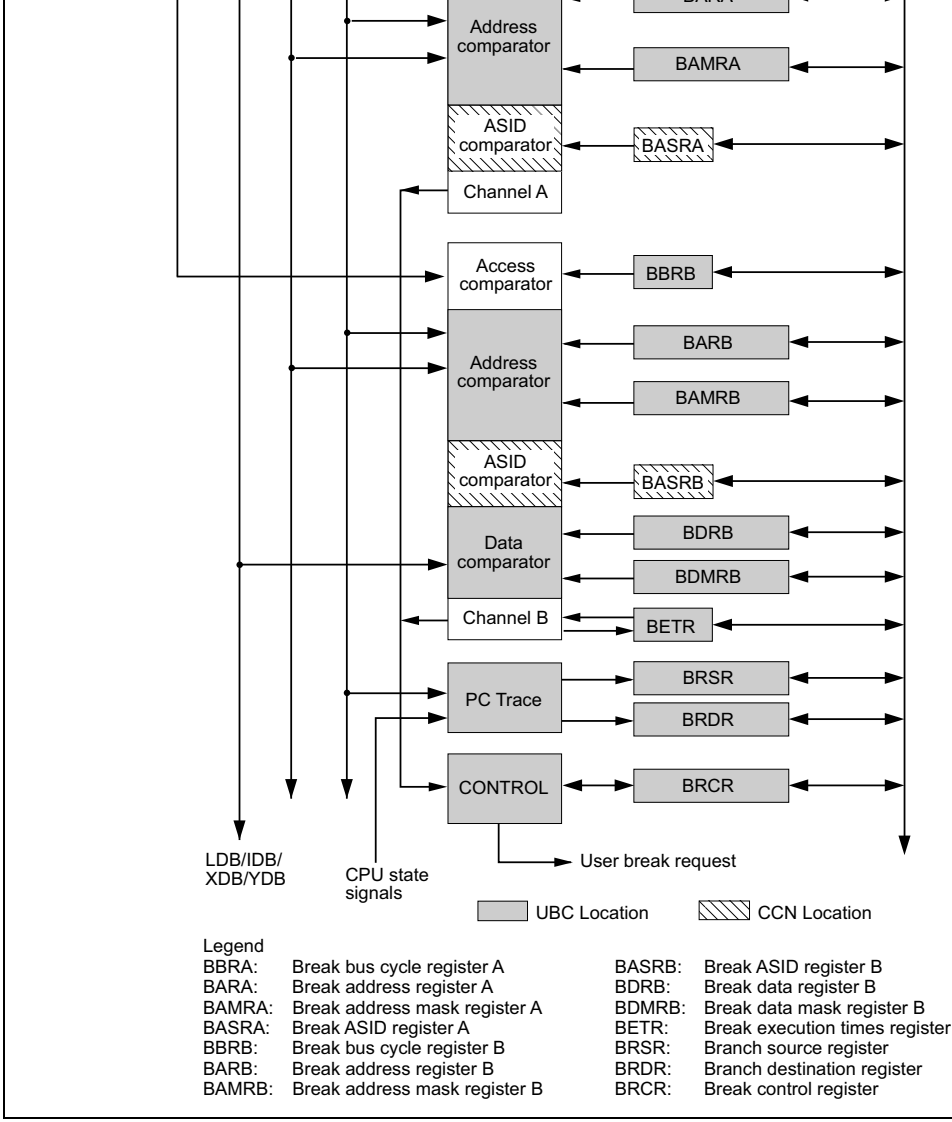


Figure 7.1 Block Diagram of User Break Controller

Break bus cycle register A	BBRA	R/W	H'0000	H'FFFFFFB8	16
Break address register B	BARB	R/W	H'00000000	H'FFFFFFA0	32
Break address mask register B	BAMRB	R/W	H'00000000	H'FFFFFFA4	32
Break bus cycle register B	BBRB	R/W	H'0000	H'FFFFFFA8	16
Break data register B	BDRB	R/W	H'00000000	H'FFFFFF90	32
Break data mask register B	BDMRB	R/W	H'00000000	H'FFFFFF94	32
Break control register	BRCR	R/W	H'00000000	H'FFFFFF98	32
Execution count break register	BETR	R/W	H'0000	H'FFFFFF9C	16
Branch source register	BRSR	R	Undefined*2	H'FFFFFFAC	32
Branch destination register	BRDR	R	Undefined*2	H'FFFFFFBC	32
Break ASID register A	BASRA	R/W	Undefined	H'FFFFFFE4	16
Break ASID register B	BASRB	R/W	Undefined	H'FFFFFFE8	16

Notes: 1. Initialized by power-on reset. Values held in standby state and undefined by resets.

2. Bit 31 of BRSR and BRDR (valid flag) is initialized by power-on resets. But are not initialized.

Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	23	22	21	20	19	18	17
	BAA23	BAA22	BAA21	BAA20	BAA19	BAA18	BAA17
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9
	BAA15	BAA14	BAA13	BAA12	BAA11	BAA10	BAA9
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1
	BAA7	BAA6	BAA5	BAA4	BAA3	BAA2	BAA1
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bits 31 to 0—Break Address A31 to A0 (BAA31 to BAA0): Stores the address on the IAB specifying break conditions of channel A.

R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit:	23	22	21	20	19	18	17
	BAMA23	BAMA22	BAMA21	BAMA20	BAMA19	BAMA18	BAMA
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit:	15	14	13	12	11	10	9
	BAMA15	BAMA14	BAMA13	BAMA12	BAMA11	BAMA10	BAMA
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit:	7	6	5	4	3	2	1
	BAMA7	BAMA6	BAMA5	BAMA4	BAMA3	BAMA2	BAMA
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bits 31 to 0—Break Address Mask Register A31 to A0 (BAMA31 to BAMA0): Sp
masked in the channel A break address bits specified by BARA (BAA31–BAA0).

Bits 31 to 0: BAMAn	**Description**
0	Break address bit BAAn of channel A is included in the break conditio (
1	Break address bit BAAn of channel A is masked and is not included i condition

n = 31 to 0

R/W:	R	R	R	R	R	R	R
Bit:	7	6	5	4	3	2	1
	CDA1	CDA0	IDA1	IDA0	RWA1	RWA0	SZA1
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bits 15 to 8—Reserved: These bits are always read as 0. The write value should alway

Bits 7 and 6—CPU Cycle/DMAC Cycle Select A (CDA1, CDA0): Selects the CPU c

DMAC cycle as the bus cycle of the channel A break condition.

Bit 7: CDA1	Bit 6: CDA0	Description
0	0	Condition comparison is not performed (In
*	1	The break condition is the CPU cycle
1	0	The break condition is the DMAC cycle

*: Don't care

Bits 5 and 4—Instruction Fetch/Data Access Select A (IDA1, IDA0): Selects the ins
fetch cycle or data access cycle as the bus cycle of the channel A break condition.

Bit 5: IDA1	Bit 4: IDA0	Description
0	0	Condition comparison is not performed (In
	1	The break condition is the instruction fetch cycle
1	0	The break condition is the data access cycle
	1	The break condition is the instruction fetch cycle or c cycle

Bits 1 and 0—Operand Size Select A (SZA1, SZA0): Selects the operand size of the
for the channel A break condition.

Bit 1: SZA1	Bit 0: SZA0	Description
0	0	The break condition does not include operand size
	1	The break condition is byte access
1	0	The break condition is word access
	1	The break condition is longword access

R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	23	22	21	20	19	18	17
	BAB23	BAB22	BAB21	BAB20	BAB19	BAB18	BAB17
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9
	BAB15	BAB14	BAB13	BAB12	BAB11	BAB10	BAB9
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1
	BAB7	BAB6	BAB5	BAB4	BAB3	BAB2	BAB1
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	23	22	21	20	19	18	17
	BAMB23	BAMB22	BAMB21	BAMB20	BAMB19	BAMB18	BAMB
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9
	BAMB15	BAMB14	BAMB13	BAMB12	BAMB11	BAMB10	BAMB
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1
	BAMB7	BAMB6	BAMB5	BAMB4	BAMB3	BAMB2	BAMB
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bits 31 to 0—Break Address Mask Register B31 to B0 (BAMB31 to BAMB0): Sp
masked in the channel B break address bits specified by BARB (BAB31—BAB0).

Bits 31 to 0: BAMBn	**Description**
0	Break address BABn of channel B is included in the break condition (
1	Break address BABn of channel B is masked and is not included in th condition

n = 31 to 0

Bit:	23	22	21	20	19	18	17
	BDB23	BDB22	BDB21	BDB20	BDB19	BDB18	BDB17
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9
	BDB15	BDB14	BDB13	BDB12	BDB11	BDB10	BDB9
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1
	BDB7	BDB6	BDB5	BDB4	BDB3	BDB2	BDB1
Initial value:	0	0	0	0	0	0	0

R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	23	22	21	20	19	18	17
	BDMB23	BDMB22	BDMB21	BDMB20	BDMB19	BDMB18	BDMB
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9
	BDMB15	BDMB14	BDMB13	BDMB12	BDMB11	BDMB10	BDMB
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1
	BDMB7	BDMB6	BDMB5	BDMB4	BDMB3	BDMB2	BDMB
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bits 31 to 0—Break Data Mask Register B31 to B0 (BDMB31 to BDMB0): Specif
the channel B break data bits specified by BDRB (BDB31—BDB0).

Bits 31 to 0:

BDMBn	Description
0	Break data BDBn of channel B is included in the break condition
1	Break data BDBn of channel B is masked and is not included in the b condition

n = 31 to 0

Notes: 1. Specify an operand size when including the value of the data bus in the bre
2. When a byte size is specified as a break condition, the same-byte data mus bits 15 to 8 and bits 7 to 0 in BDRB for the break data.

R/W:	R	R	R	R	R	R	R
Bit:	7	6	5	4	3	2	1
	CDB1	CDB0	IDB1	IDB0	RWB1	RWB0	SZB1
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bits 15 to 8—Reserved: These bits are always read as 0. These bits are always read as

Bits 7 and 6—CPU Cycle/DMAC Cycle Select B (CDB1, CDB0): Select the CPU cy
DMAC cycle as the bus cycle of the channel B break condition.

Bit 7: CDB1	Bit 6: CDB0	Description
0	0	Condition comparison is not performed (I
*	1	The break condition is the CPU cycle
1	0	The break condition is the DMAC cycle

*: Don't care

Bits 5 and 4—Instruction Fetch/Data Access Select B (IDB1, IDB0): Select the instr
fetch cycle or data access cycle as the bus cycle of the channel B break condition.

Bit 5: IDB1	Bit 4: IDB0	Description
0	0	Condition comparison is not performed (I
	1	The break condition is the instruction fetch cycle
1	0	The break condition is the data access cycle
	1	The break condition is the instruction fetch cycle or d cycle

Bits 1 and 0—Operand Size Select B (SZB1, SZB0): Select the operand size of the
the channel B break condition.

Bit 1: SZB1	Bit 0: SZB0	Description
0	0	The break condition does not include operand size
	1	The break condition is byte access
1	0	The break condition is word access
	1	The break condition is longword access

4. Determine whether to include data bus on channel B in comparison conditions.
5. Enable PC trace.
6. Enable the ASID check.

The break control register (BRCR) is a 32-bit read/write register that has break conditio flags and bits for setting a variety of break conditions. A power-on reset initializes BR H'00000000.

Bit:	31	30	29	28	27	26	25
	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R

Bit:	23	22	21	20	19	18	17
	—	—	BASMA	BASMB	—	—	—
Initial value:	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R	R	R

Bit:	15	14	13	12	11	10	9
	SCMFCA	SCMFCB	SCMFDA	SCMFDB	PCTE	PCBA	—
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R

Bit:	7	6	5	4	3	2	1
	DBEB	PCBB	—	—	SEQ	—	—
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R	R	R/W	R	R

Bits 31 to 22—Reserved: These bits are always read as 0. The write value should alwa

ASID0 (BASB7 to BASB0) set in BASRB are masked or not.

Bit 20: BASMB	Description
0	All BASRB bits are included in break condition, ASID is checked
1	No BASRB bits are included in break condition, ASID is not checked

Bits 19 to 16—Reserved: These bits are always read as 0. The write value should alw

Bit 15—CPU Condition Match Flag A (SCMFCA): When the CPU bus cycle condi break conditions set for channel A is satisfied, this flag is set to 1 (not cleared to 0). In clear this flag, write 0 into this bit.

Bit 15: SCMFCA	Description
0	The CPU cycle condition for channel A does not match
1	The CPU cycle condition for channel A matches

Bit 14—CPU Condition Match Flag B (SCMFCB): When the CPU bus cycle condi break conditions set for channel B is satisfied, this flag is set to 1 (not cleared to 0). In clear this flag, write 0 into this bit.

Bit 14: SCMFCB	Description
0	The CPU cycle condition for channel B does not match
1	The CPU cycle condition for channel B matches

Bit 12—DMAC Condition Match Flag B (SCMFDB): When the on-chip DMAC bus condition in the break conditions set for channel B is satisfied, this flag is set to 1 (not c 0). In order to clear this flag, write 0 into this bit.

Bit 12: SCMFDB	Description	
0	The DMAC cycle condition for channel B does not match	(In
1	The DMAC cycle condition for channel B matches	

Bit 11—PC Trace Enable (PCTE): Enables PC trace.

Bit 11: PCTE	Description	
0	Disables PC trace	(In
1	Enables PC trace	

Bit 10—PC Break Select A (PCBA): Selects the break timing of the instruction fetch channel A as before or after instruction execution.

Bit 10: PCBA	Description	
0	PC break of channel A is set before instruction execution	(In
1	PC break of channel A is set after instruction execution	

Bits 9 and 8—Reserved: These bits are always read as 0. The write value should alway

Bit 7—Data Break Enable B (DBEB): Selects whether or not the data bus condition i in the break condition of channel B.

Bit 7: DBEB	Description	
0	No data bus condition is included in the condition of channel B	(In
1	The data bus condition is included in the condition of channel B	

Bit 3—Sequence Condition Select (SEQ): Selects two conditions of channels A and independent or sequential.

Bit 3: SEQ	Description
0	Channels A and B are compared under the independent condition (
1	Channels A and B are compared under the sequential condition (cha channel B)

Bits 2 and 1—Reserved: These bits are always read as 0. The write value should alw

Bit 0—The Number of Execution Times Break Enable (ETBE): Enable the execut break condition only on channel B. If this bit is 1 (break enable), a user break is issue number of break conditions matches with the number of execution times that is specifi BETR register.

Bit 0: ETBE	Description
0	The execution-times break condition is masked on channel B (
1	The execution-times break condition is enabled on channel B

	—	—	—	—			
Initial value:	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	31	30	29	28	27	26	25
	SVF	PID2	PID1	PID0	BSA27	BSA26	BSA2
Initial value:	0	*	*	*	*	*	*
R/W:	R	R	R	R	R	R	R

Bit:	23	22	21	20	19	18	17
	BSA23	BSA22	BSA21	BSA20	BSA19	BSA18	BSA1
Initial value:	*	*	*	*	*	*	*
R/W:	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9
	BSA15	BSA14	BSA13	BSA12	BSA11	BSA10	BSA
Initial value:	*	*	*	*	*	*	*
R/W:	R	R	R	R	R	R	R

Bit:	7	6	5	4	3	2	1
	BSA7	BSA6	BSA5	BSA4	BSA3	BSA2	BSA
Initial value:	*	*	*	*	*	*	*
R/W:	R	R	R	R	R	R	R

Note: * Undefined value

Bit 31—BRSR Valid Flag (SVF): Indicates whether the address and the pointer by w branch source address can be calculated. When a branch source address is fetched, this to 1. This flag is cleared to 0 in reading BRSR.

Bit 31: SVF	Description
0	The value of BRSR register is invalid
1	The value of BRSR register is valid

Bits 27 to 0—Branch Source Address (BSA27 to BSA0): These bits store the last fet address before branch.

7.2.12 Branch Destination Register (BRDR)

BRDR is a 32-bit read register. BRDR stores the branch destination fetch address. BRD flag bit that is set to 1 when branch occurs. This flag bit is cleared to 0, when BRDR is also initialized by power-on resets or manual resets. Other bits are not initialized by res BRDR registers have queue structure and a stored register is shifted every branch.

Bit:	31	30	29	28	27	26	25
	DVF	—	—	—	BDA27	BDA26	BDA25
Initial value:	0	*	*	*	*	*	*
R/W:	R	R	R	R	R	R	R

Bit:	23	22	21	20	19	18	17
	BDA23	BDA22	BDA21	BDA20	BDA19	BDA18	BDA17
Initial value:	*	*	*	*	*	*	*
R/W:	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9
	BDA15	BDA14	BDA13	BDA12	BDA11	BDA10	BDA9
Initial value:	*	*	*	*	*	*	*
R/W:	R	R	R	R	R	R	R

Bit:	7	6	5	4	3	2	1
	BDA7	BDA6	BDA5	BDA4	BDA3	BDA2	BDA1
Initial value:	*	*	*	*	*	*	*
R/W:	R	R	R	R	R	R	R

Note: * Undefined value

Bits 30 to 28—Reserved: These bits are always read as 0. The write value should alw

Bits 27 to 0—Branch Destination Address (BDA27 to BDA0): These bits store the address after branch.

7.2.13 Break ASID Register A (BASRA)

Break ASID register A (BASRA) is an 8-bit read/write register that specifies the ASID as the break condition for channel A. It is not initialized by resets.

Bit:	7	6	5	4	3	2	1
	BASA7	BASA6	BASA5	BASA4	BASA3	BASA2	BASA
Initial value:	*	*	*	*	*	*	*
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: * Undefined value

Bits 7 to 0—Break ASID A7 to 0 (BASA7 to BASA0): These bits store the ASID (b that is the channel A break condition.

7.2.14 Break ASID Register B (BASRB)

Break ASID register B (BASRB) is an 8-bit read/write register that specifies the ASID as the break condition for channel B. It is not initialized by resets.

Bit:	7	6	5	4	3	2	1
	BASB7	BASB6	BASB5	BASB4	BASB3	BASB2	BASB
Initial value:	*	*	*	*	*	*	*
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: * Undefined value

Bits 7 to 0—Break ASID A7 to 0 (BASB7 to BASB0): These bits store the ASID (bi that is the channel B break condition.

break data register (BDRB). The masked data is set in the break data mask register (
The breaking bus conditions are set in the break bus cycle registers (BBRA and BB
groups of the BBRA and BBRB (CPU cycle/DMAC cycle select, instruction fetch/d
select, and read/write select) are each set. No user break will be generated if even on
groups is set with 00. The respective conditions are set in the bits of the BRCR.

2. When the break conditions are satisfied, the UBC sends a user break request to the i
controller. The break type will be sent to CPU indicating the instruction fetch, pre/p
instruction break, data access break. When conditions match up, the CPU condition
flags (SCMFCA and SCMFCB) and DMAC condition match flags (SCMFDA and
for the respective channels are set.

3. The appropriate condition match flags (SCMFCA, SCMFDA, SCMFCB, and SCM
be used to check if the set conditions match or not. The matching of the conditions s
but they are not reset. 0 must first be written to them before they can be used again.

4. There is a chance that the data access break and its following instruction fetch break
around the same time, there will be only one break request to the CPU, but these tw
channel match flags could be both set.

7.3.2 Break on Instruction Fetch Cycle

1. When CPU/instruction fetch/read/word or longword is set in the break bus cycle reg
(BBRA/BBRB), the break condition becomes the CPU instruction fetch cycle. Whe
breaks before or after the execution of the instruction can then be selected with the
PCBA/PCBB bits of the break control register (BRCR) for the appropriate channel.

2. An instruction set for a break before execution breaks when it is confirmed that the
has been fetched and will be executed. This means this feature cannot be used on in
fetched by overrun (instructions fetched at a branch or during an interrupt transition
be executed). When this kind of break is set for the delay slot of a delay branch instr
break is generated prior to execution of the instruction that then first accepts the bre
Meanwhile, the break set for pre-instruction-break on delay slot instruction and post
instruction-break on SLEEP instruction are also prohibited.

7.3.3 Break by Data Access Cycle

1. The memory cycles in which CPU data access breaks occur are from instructions.

2. The relationship between the data access cycle address and the comparison conditi
   operand size are listed in table 7.2:

Table 7.2 Data Access Cycle Addresses and Operand Size Comparison Condit

Access Size	Address Compared
Longword	Compares break address register bits 31–2 to address bus bits
Word	Compares break address register bits 31–1 to address bus bits
Byte	Compares break address register bits 31–0 to address bus bits

This means that when address H'00001003 is set without specifying the size condi
example, the bus cycle in which the break condition is satisfied is as follows (wher
conditions are met).

Longword access at H'00001000

Word access at H'00001002

Byte access at H'00001003

3. When the data value is included in the break conditions on B channel:

   When the data value is included in the break conditions, either longword, word, or
   specified as the operand size of the break bus cycle registers (BBRA and BBRB).
   values are included in break conditions, a break is generated when the address con
   data conditions both match. To specify byte data for this case, set the same data in
   bits 15–8 and bits 7–0 of the break data register B (BDRB) and break data mask re
   (BDMRB). When word or byte is set, bits 31–16 of BDRB and BDMRB are ignor

4. When the DMAC data access is included in the break condition:

   When the address is included in the break condition on DMAC data access, the op
   the break bus cycle registers (BBRA and BBRB) should be byte, word or no specif
   size. When the data value is included, select either byte or word.

condition is specified, the break condition is satisfied at channel B condition match
= H'0001 after channel A condition match.

7.3.5 Value of Saved Program Counter

The PC when a break occurs is saved to the SPC in user breaks. The PC value saved is
depending on the type of break.

1. When instruction fetch (before instruction execution) is specified as a break conditi
   The value of the program counter (PC) saved is the address of the instruction that m
   break condition. The fetched instruction is not executed, and a break occurs before i

2. When instruction fetch (after instruction execution) is specified as a break condition
   The PC value saved is the address of the instruction to be executed following the ins
   which the break condition matches. The fetched instruction is executed, and a break
   before the execution of the next instruction.

3. When data access (address only) is specified as a break condition:
   The PC value is the address of the instruction to be executed following the instructi
   matched the break condition. The instruction that matched the condition is executed
   break occurs before the next instruction is executed.

4. When data access (address + data) is specified as a break condition:
   The PC value is the start address of the instruction that follows the instruction alread
   when break processing started up. When a data value is added to the break condition
   place where the break will occur cannot be specified exactly. The break will occur b
   execution of an instruction fetched around the data access where the break occurred

and IA (the instruction address before branch occurs) is as follows: IA = BSA – 2

Notes are needed when an interrupt (a branch) is issued before the branch destinati instruction is executed. In case of the next figure, the instruction “Exec” executed before branch is calculated by IA = BSA – 2 * PID. However, when branch “branc slot and the destination address is 4n + 2 address, the address “Dest” which is spec branch instruction is stored in BRSR (Dest = BSA). Therefore, as IA = BSA – 2 * applied to this case, this PID is invalid. The case where BSA is 4n + 2 boundary is only to this case and then some cases are classified as follows:

```
Exec:branch  Dest
Dest:instr        (not executed)
      interrupt
Int: interrupt routine
```

If the PID value is odd, instruction buffer indicates PID+2 buffer. However, these in this table are accounted for it. Therefore, the true branch source address is calcu BSA and PID values stored in BRSR.

3. The branch address before branch occurrence, IA, has different values due to some branch.
   a. Branch instruction

      The branch instruction address
   b. Interrupt

      The last instruction executed before interrupt

      The top address of interrupt routine is stored in BRDR.

4. BRSR and BRDR have eight pairs of queue structures. The top of queues is read fi address stored in the PC trace register is read. BRSR and BRDR share the read poi BRSR and BRDR in order, the queue only shifts after BRDR is read. When readin longword access should be used. Also, the PC trace has a trace pointer, which initi to the bottom of the queues. The first pair of branch addresses will be stored at the the queues, then push up when next pairs come into the queues. The trace pointer v the next branch address to be executed, unless it got push out of the queues. When address has been executed, the trace pointer will shift down to next pair of address

BARA = H'00000404, BAMRA = H'00000000, BBRA = H'0054, BARB = H'00008
BAMRB = H'00000006, BBRB = H'0054, BDRB = H'00000000, BDMRB = H'000
BRCR = H'00300400

Specified conditions: Channel A/channel B independent mode

- Channel A

  Address: H'00000404, Address mask: H'00000000

  Bus cycle: CPU/instruction fetch (after instruction execution)/read (operand si
  included in the condition)

  No ASID check is included

- Channel B

  Address: H'00008010, Address mask: H'00000006

  Data: H'00000000, Data mask: H'00000000

  Bus cycle: CPU/instruction fetch (before instruction execution)/read (operand
  included in the condition)

  No ASID check is included

A user break occurs after an instruction of address H'00000404 is executed or befor
instructions of adresses H'00008010 to H'00008016 are executed.

- Channel B
  Address: H'0003722E, Address mask: H'00000000, ASID = H'70
  Data: H'00000000, Data mask: H'00000000
  Bus cycle: CPU/instruction fetch (before instruction execution)/read/word

An instruction with ASID = H'80 and address H'00037226 is executed, and a user
before an instruction with ASID = H'70 and address H'0003722E is executed.

3. Register specifications
   BARA = H'00027128, BAMRA = H'00000000, BBRA = H'005A, BARB = H'000
   BAMRB = H'00000000, BBRB = H'0054, BDRB = H'00000000, BDMRB = H'00
   BRCR = H'00300000

   Specified conditions: Channel A/channel B independent mode
   - Channel A
     Address: H'00027128, Address mask: H'00000000
     Bus cycle: CPU/instruction fetch (before instruction execution)/write/word
     No ASID check is included
   - Channel B
     Address: H'00031415, Address mask: H'00000000
     Data: H'00000000, Data mask: H'00000000
     Bus cycle: CPU/instruction fetch (before instruction execution)/read (operanc
     included in the condition)
     No ASID check is included

   On channel A, no user break occurs since instruction fetch is not a write cycle. On
   no user break occurs since instruction fetch is performed for an even address.

- Channel B
  Address: H'0003722E, Address mask: H'00000000, ASID: H'70
  Data: H'00000000, Data mask: H'00000000
  Bus cycle: CPU/instruction fetch (before instruction execution)/read/word

Since instruction fetch is not a write cycle on channel A, a sequence condition does
Therefore, no user break occurs.

5. Register specifications
   BARA = H'00000500, BAMRA = H'00000000, BBRA = H'0057, BARB = H'0000
   BAMRB = H'00000000, BBRB = H'0057, BDRB = H'00000000, BDMRB = H'000
   BRCR = H'00300001, BETR = H'0005

   Specified conditions: Channel A/channel B independent mode
   - Channel A
     Address: H'00000500, Address mask: H'00000000
     Bus cycle: CPU/instruction fetch (before instruction execution)/read/longword
   - Channel B
     Address: H'00001000, Address mask: H'00000000
     Data: H'00000000, Data mask: H'00000000
     Bus cycle: CPU/instruction fetch (before instruction execution)/read/longword
     The number of execution-times break enable (5 times)

   On channel A, a user break occurs before an instruction of address H'00000500 is e
   On channel B, a user break occurs before the fifth instruction execution after instruc
   address H'00001000 are executed four times.

Bus cycle: CPU/instruction fetch (after instruction execution)/read (operand s
included in the condition)

- Channel B
  Address: H'00008010, Address mask: H'00000006, ASID: H'70
  Data: H'00000000, Data mask: H'00000000
  Bus cycle: CPU/instruction fetch (before instruction execution)/read (operand
  included in the condition)

A user break occurs after an instruction with ASID = H'80 and address H'0000800
H'00008FFE is executed or before instructions with ASID = H'70 and addresses H
to H'00008016 are executed.

Break Condition Specified to a CPU Data Access Cycle

1. Register specifications
   BARA = H'00123456, BAMRA = H'00000000, BBRA = H'0064, BARB = H'000A
   BAMRB = H'000000FF, BBRB = H'006A, BDRB = H'0000A512, BDMRB = H'0
   BRCR = H'00000080, BASRA = H'80, BASRB = H'70

   Specified conditions: Channel A/channel B independent mode

   - Channel A
     Address: H'00123456, Address mask: H'00000000
     Bus cycle: CPU/data access/read (operand size is not included in the condition
   - Channel B
     Address: H'000ABCDE, Address mask: H'000000FF, ASID: H'70
     Data: H'0000A512, Data mask: H'00000000
     Bus cycle: CPU/data access/write/word

   On channel A, a user break occurs with ASID = H'80 during longword read to add
   H'00123454, word read to address H'00123456, or byte read to address H'0012345
   channel B, a user break occurs with ASID = H'70 when word H'A512 is written in
   H'000ABC00 to H'000ABCFE.

Address: H'00314156, Address mask: H'00000000, ASID: H'80

Bus cycle: DMAC/instruction fetch/read (operand size is not included in the co

- Channel B

  Address: H'00055555, Address mask: H'00000000, ASID: H'70

  Data: H'00000078, Data mask: H'0000000F

  Bus cycle: DMAC/data access/write/byte

On channel A, no user break occurs since instruction fetch is not performed in DMA
On channel B, a user break occurs with ASID = H'70 when the DMAC writes byte l
address H'00055555.

occur simultaneously, is set.

b. Since the CPU has a pipeline configuration, the pipeline determines the order o instruction fetch cycle and a memory cycle. Therefore, when a channel conditio in the order of bus cycles, a sequential condition is satisfied.

c. When the bus cycle condition for channel A is specified as a break before exec (PCBA = 0 in BRCR) and an instruction fetch cycle (in BBRA), the attention i A break is issued and condition match flags in BRCR are set to 1, when the bus conditions both for channels A and B match simultaneously.

4. The change of a UBC register value is executed in MA (memory access) stage. Th even if the break condition matches in the instruction fetch address following the i which the pre-execution break is specified as the break condition, no break occurs. know the timing UBC register is changed, read the last written register. Instruction are valid for the newly written register value.
5. The branch instruction should not be executed as soon as PC trace register BRSR a are read.
6. When PC breaks and TLB exceptions or errors occur in the same instruction. The follows:

   a. Break and instruction fetch exceptions: Instruction fetch exception occurs first.

   b. Break before execution and operand exception: Break before execution occurs

   c. Break after execution and operand exception: Operand exception occurs first.

The SH7709S has the following power-down modes and function:

1. Sleep mode
2. Standby mode
3. Module standby function (TMU, RTC, SCI, UBC, DMAC, DAC, ADC, SCIF, and chip peripheral modules)
4. Hardware standby mode

Table 8.1 shows the transition conditions for entering the modes from the program exe as well as the CPU and peripheral module states in each mode and the procedures for e each mode.

Standby mode	Execute SLEEP instruction with STBY bit set to 1 in STBCR	Halts	Halts	Held	Held	Halt[*1]	Held	Self-refresh	1. I 2. F
Module standby function	Set MSTP bit to 1 in STBCR	Runs	Runs or halts	Held	Held	Specified module halts	[*2]	Refresh	1. C b 2. F
Hardware standby mode	Drive CA pin low	Halts	Halts	Held	Held	Halt[*3]	Held	Self-refresh	Pov

Notes: 1. The RTC still runs if the START bit in RCR2 is set to 1 (see section 13, Real (RTC)). The TMU still runs when output of the RTC is used as input to its co section 12, Timer (TMU)).

2. Depends on the on-chip peripheral module.
   TMU external pin: Held
   SCI external pin: Reset
3. The RTC still runs if the START bit in RCR2 is set to 1. The TMU does not ru

			LL: Normal operation
Wakeup from standby mode	$\overline{\text{WAKEUP}}$	O	Active-low assertion after accepting wa interrupt in standby mode until returning operation with WDT overflow

Note: H: high level; L: low level

8.1.3 Register Configuration

Table 8.3 shows the control register configuration for the power-down modes.

Table 8.3 Register Configuration

Name	Abbreviation	R/W	Initial Value	Access Size
Standby control register	STBCR	R/W	H'00*	H'FFFFFF82
Standby control register 2	STBCR2	R/W	H'00*	H'FFFFFF88

Note: * Initialized by a power-on reset. This value is not initialized by a manual reset; value is retained.

8.2 Register Descriptions

8.2.1 Standby Control Register (STBCR)

The standby control register (STBCR) is an 8-bit readable/writable register that sets th down mode. STBCR is initialized to H'00 by a power-on reset.

Bit:	7	6	5	4	3	2	1
	STBY	—	—	STBXTL	—	MSTP2	MSTP
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R	R	R/W	R	R/W	R/W

in standby mode.

Bit 4: STBXTL	Description	
0	Clock pulse generator is halted in standby mode	(I
1	Clock pulse generator is operates in standby mode	

Bit 2—Module Standby 2 (MSTP2): Specifies halting of the clock supply to the timer (an on-chip peripheral module). When the MSTP2 bit is set to 1, the supply of the clock TMU is halted.

Bit 2: MSTP2	Description	
0	TMU runs	(I
1	Clock supply to TMU is halted	

Bit 1—Module Standby 1 (MSTP1): Specifies halting of the clock supply to the realti RTC (an on-chip peripheral module). When the MSTP1 bit is set to 1, the supply of the the RTC is halted. When the clock halts, all RTC registers become inaccessible, but the keeps running.

Bit 1: MSTP1	Description	
0	RTC runs	(I
1	Clock supply to RTC is halted	

Before switching the RTC to module standby, access at least one among the registers R and TMU.

8.2.2 Standby Control Register 2 (STBCR2)

The standby control register 2 (STBCR2) is a readable/writable 8-bit register that sets down mode. STBCR2 is initialized to H'00 by a power-on reset.

Bit:	7	6	5	4	3	2	1
	—	MDCHG	MSTP8	MSTP7	MSTP6	MSTP5	MST
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7—Reserved: The write value set in the program should always be 1.

Bit 6—Pin MD5 to MD0 Control (MDCHG): Specifies whether or not pins MD5 to changed in standby mode. When this bit is set to 1, the MD5 to MD0 pin values are la returning from standby mode by means of a reset or interrupt.

Bit 6: MDCHG	Description
0	Pins MD5 to MD0 are not changed in standby mode
1	Pins MD5 to MD0 are changed in standby mode

Bit 5— Module Stop 8 (MSTP8): Specifies halting of the clock supply to the user br controller UBC (an on-chip peripheral module). When the MSTP8 bit is set to 1, the s clock to the UBC is halted.

Bit 5: MSTP8	Description
0	UBC runs
1	Clock supply to UBC is halted

Bit 3—Module Stop 6 (MSTP6): Specifies halting of the clock supply to the DAC (an peripheral module). When the MSTP6 bit is set to 1, the supply of the clock to the DAC

Bit 3: MSTP6	Description
0	DAC runs (I
1	Clock supply to DAC halted

Bit 2—Module Stop 5 (MSTP5): Specifies halting of the clock supply to the ADC (an peripheral module). When the MSTP5 bit is set to 1, the supply of the clock to the ADC and all registers are initialized.

Bit 2: MSTP5	Description
0	ADC runs (I
1	Clock supply to ADC halted and all registers initialized

Bit 1—Module Stop 4 (MSTP4): Specifies halting of the clock supply to the SCI2 (SC communication interface with FIFO (an on-chip peripheral module). When the MSTP1 1, the supply of the clock to SCI2 (SCIF) is halted.

Bit 1: MSTP4	Description
0	SCI2 (SCIF) runs (I
1	Clock supply to SCI2 (SCIF) halted

Bit 0—Module Stop 3 (MSTP3): Specifies halting of the clock supply to the SCI1 (Ir Infrared Data Association interface with FIFO (an on-chip peripheral module). When th bit is set to 1, the supply of the clock to SCI1 (IrDA) is halted.

Bit 0: MSTP3	Description
0	SCI1(IrDA) runs (I
1	Clock supply to SCI1(IrDA) halted

CKIO2 pins. In sleep mode, the STATUS1 pin is set high and the STATUS0 pin low.

8.3.2 Canceling Sleep Mode

Sleep mode is canceled by an interrupt (NMI, IRQ, IRL, on-chip peripheral module, P reset. Interrupts are accepted in sleep mode even when the BL bit in the SR register is necessary, save SPC and SSR to the stack before executing the SLEEP instruction.

Canceling with an Interrupt: When an NMI, IRQ, IRL or on-chip peripheral module occurs, sleep mode is canceled and interrupt exception handling is executed. A code in interrupt source is set in the INTEVT and INTEVT2 registers.

Canceling with a Reset: Sleep mode is canceled by a power-on reset or a manual res

8.3.3 Precautions when Using the Sleep Mode

DMAC transfers should not be performed in the sleep mode under conditions other tha clock ratio of Iϕ (on-chip clock) to Bϕ (bus clock) is 1:1.

states of registers in standby mode.

Table 8.4 Register States in Standby Mode

Module	Registers Initialized	Registers Retaining D
Interrupt controller (INTC)	—	All registers
On-chip clock pulse generator (OSC)	—	All registers
User break controller (UBC)	—	All registers
Bus state controller (BSC)	—	All registers
Timer unit (TMU)	TSTR register	Registers other than T
Realtime clock (RTC)	—	All registers
A/D converter (ADC)	All registers	—
D/A converter (DAC)	—	All registers

The procedure for moving to standby mode is as follows:

1. Clear the TME bit in the WDT's timer control register (WTCSR) to 0 to stop the W the WDT's timer counter (WTCNT) to 0 and the CKS2–CKS0 bits in the WTCSR r appropriate values to secure the specified oscillation settling time.
2. After the STBY bit in the STBCR register is set to 1, a SLEEP instruction is execut
3. Standby mode is entered and the clocks within the chip are halted. The STATUS1 p goes low and the STATUS0 pin output goes high.

Interrupt handling then begins and a code indicating the interrupt source is set in the IN
INTEVT2 registers. After the branch to the interrupt handling routine, clear the STBY
STBCR register. WTCNT stops automatically. If the STBY bit is not cleared, WTCNT
operation and a transition is made to standby mode*3 when it reaches H'80. This funct
the data from being destroyed due to a rise in voltage with an unstable power supply, e
Interrupts are accepted in standby mode even when the BL bit in the SR register is 1. I
save SPC and SSR to the stack before executing the SLEEP instruction. Immediately a
interrupt is detected, the phase of the CKIO pin clock output may be unstable, until the
starts interrupt handling. (The canceling condition is that the IRL3–IRL0 level is high
mask level in the I3–I0 bits in the SR register.)

Notes: 1. When the RTC is being used, standby mode can be canceled using IRL3–I
IRQ0, or PINT0/1.
2. Standby mode can be canceled with an RTC or TMU (only when running o
clock) interrupt.
3. This standby mode can be canceled only by a power-on reset.

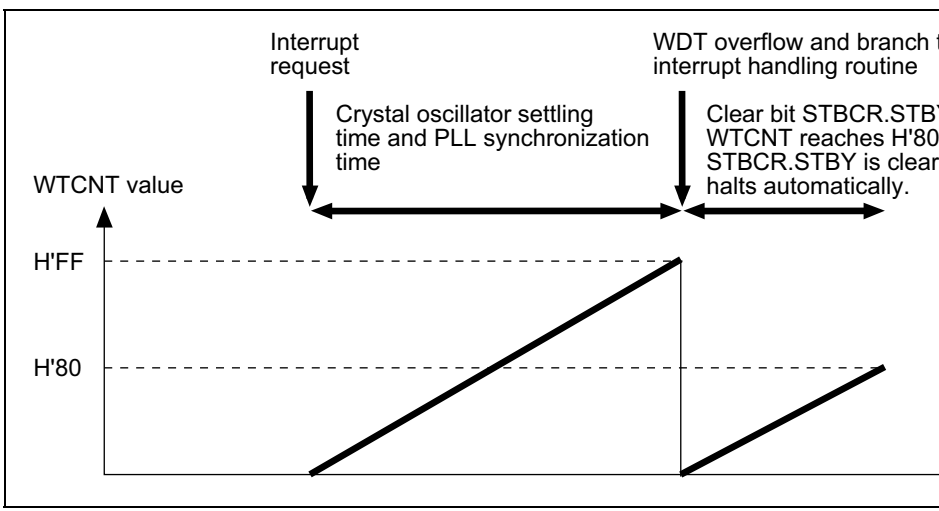


Figure 8.1 Canceling Standby Mode with STBCR.STBY

1. Enter standby mode using the appropriate procedures.
2. Once standby mode is entered and the clock stopped within the chip, the STATUS1
is low and the STATUS0 pin output is high.
3. Once the STATUS1 pin goes low and the STATUS0 pin goes high, the input clock
or the frequency is changed.
4. When the frequency is changed, an NMI, IRL, IRQ, PINT, or on-chip peripheral mo
(except interval timer) interrupt is input after the change. When the clock is stopped
interrupts are input after the clock is applied.
5. After the time set in the WDT has elapsed, the clock starts being applied internally
chip, the STATUS1 and STATUS0 pins both go low, and operation resumes from i
exception handling.

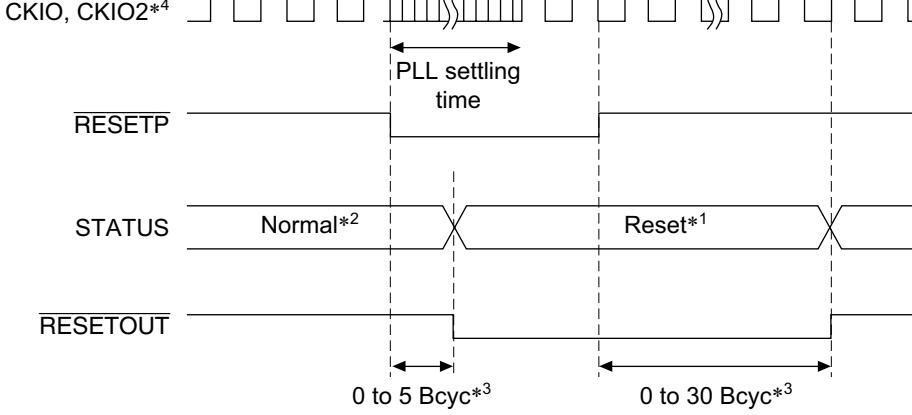
their values.

Bit	Value	Description
MSTP8	0	UBC runs
	1	Supply of clock to UBC halted
MSTP7	0	DMAC runs
	1	Supply of clock to DMAC halted
MSTP6	0	DAC runs
	1	Supply of clock to DAC halted
MSTP5	0	ADC runs
	1	Supply of clock to ADC halted, and all registers initialized
MSTP4	0	SCIF runs
	1	Supply of clock to SCIF halted
MSTP3	0	IrDA runs
	1	Supply of clock to IrDA halted
MSTP2	0	TMU runs
	1	Supply of clock to TMU halted. Registers initialized*1
MSTP1	0	RTC runs
	1	Supply of clock to RTC halted. Register access prohibited*2*3
MSTP0	0	SCI runs
	1	Supply of clock to SCI halted

Notes: 1. The registers initialized are the same as in standby mode (see table 8.4).
2. The counter runs.
3. Before switching the RTC to module standby, access at least one among th
RTC, SCI, and TMU.

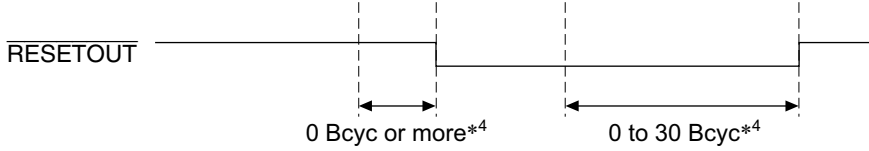
8.5.2 Clearing Module Standby Function

The module standby function can be cleared by clearing the MSTPSLP0 and MSTP8–
to 0, or by a power-on reset or manual reset.



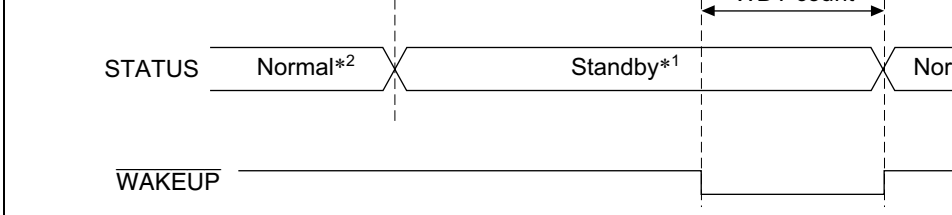
Notes: 1. Reset: HH (STATUS1 high, STATUS0 high)
2. Normal: LL (STATUS1 low, STATUS0 low)
3. Bcyc: Bus clock cycle
4. The CKIO2 output is available only in clock modes 0, 1, and 2.

Figure 8.2 Power-On Reset (Clock Modes 0, 1, 2, and 7) STATUS Outpu



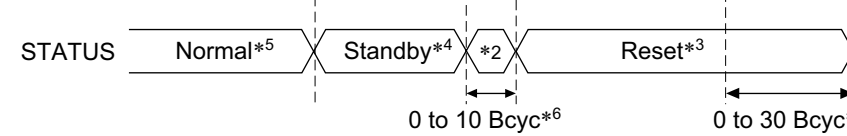
Notes: 1. In a manual reset, STATUS becomes HH (reset) and the internal rese
after waiting for the executing bus cycle to end.
2. Reset: HH (STATUS1 high, STATUS0 high)
3. Normal: LL (STATUS1 low, STATUS0 low)
4. Bcyc: Bus clock cycle
5. The CKIO2 output is available only in clock modes 0, 1, and 2.

Figure 8.3 Manual Reset STATUS Output



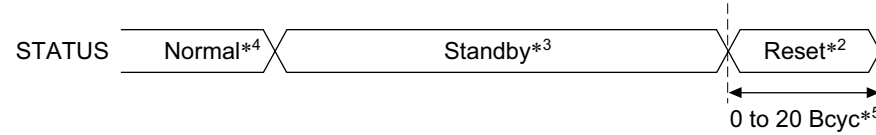
Notes: 1. Standby: LH (STATUS1 low, STATUS0 high)
2. Normal: LL (STATUS1 low, STATUS0 low)
3. The CKIO2 output is available only in clock modes 0, 1, and 2.

Figure 8.4 Standby to Interrupt STATUS Output



Notes: 1. When standby mode is cleared with a power-on reset, the WDT doe
count. Keep $\overline{\text{RESETP}}$ low during the PLL's oscillation settling time.
2. Undefined
3. Reset: HH (STATUS1 high, STATUS0 high)
4. Standby: LH (STATUS1 low, STATUS0 high)
5. Normal: LL (STATUS1 low, STATUS0 low)
6. Bcyc: Bus clock cycle
7. The CKIO2 output is available only in clock modes 0, 1, and 2.

Figure 8.5 Standby to Power-On Reset STATUS Output

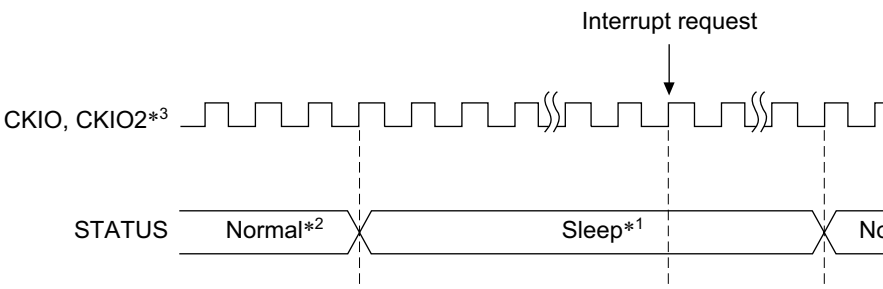


Notes: 1. When standby mode is cleared with a manual reset, the WDT does not
Keep $\overline{\text{RESETM}}$ low during the PLL's oscillation settling time.
2. Reset: HH (STATUS1 high, STATUS0 high)
3. Standby: LH (STATUS1 low, STATUS0 high)
4. Normal: LL (STATUS1 low, STATUS0 low)
5. Bcyc: Bus clock cycle
6. The CKIO2 output is available only in clock modes 0, 1, and 2.

Figure 8.6 Standby to Manual Reset STATUS Output

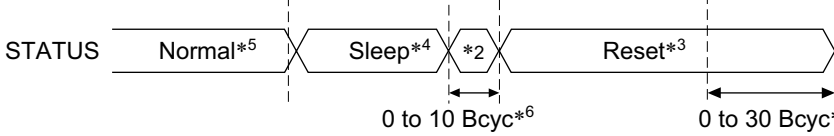
8.6.3 Timing for Canceling Sleep Mode

Sleep to Interrupt



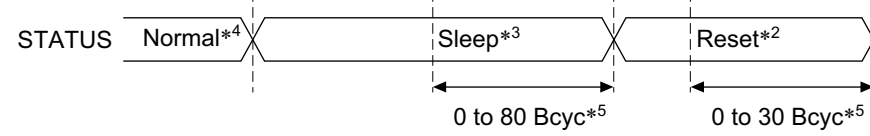
Notes: 1. Sleep: HL (STATUS1 high, STATUS0 low)
2. Normal: LL (STATUS1 low, STATUS0 low)
3. The CKIO2 output is available only in clock modes 0, 1, and 2.

Figure 8.7 Sleep to Interrupt STATUS Output



Notes: 1. When the PLL1's multiplication ratio is changed by a power-on reset keep $\overline{\text{RESETP}}$ low during the PLL's oscillation settling time.
2. Undefined
3. Reset: HH (STATUS1 high, STATUS0 high)
4. Sleep: HL (STATUS1 high, STATUS0 low)
5. Normal: LL (STATUS1 low, STATUS0 low)
6. Bcyc: Bus clock cycle
7. The CKIO2 output is available only in clock modes 0, 1, and 2.

Figure 8.8 Sleep to Power-On Reset STATUS Output



Notes: 1. Keep $\overline{\text{RESETM}}$ low until STATUS becomes reset.
2. Reset: HH (STATUS1 high, STATUS0 high)
3. Sleep: HL (STATUS1 high, STATUS0 low)
4. Normal: LL (STATUS1 low, STATUS0 low)
5. Bcyc: Bus clock cycle
6. The CKIO2 output is available only in clock modes 0, 1, and 2.

Figure 8.9 Sleep to Manual Reset STATUS Output

Hardware standby mode differs from (software) standby mode as follows:

1. Interrupts and manual resets are not accepted.
2. The TMU does not operate.

Operation when a low-level signal is input at the CA pin depends on the CPG state, as

1. In standby mode

   The clock remains stopped and the chip enters the hardware standby state. Accept interrupts and manual resets is disabled, TCLK output is fixed low, and the TMU l
2. During WDT operation when standby mode is canceled by an interrupt

   The chip enters hardware standby mode after standby mode is canceled and the CP operation.
3. In sleep mode

   The chip enters hardware standby mode after sleep mode is canceled and the CPU operation.

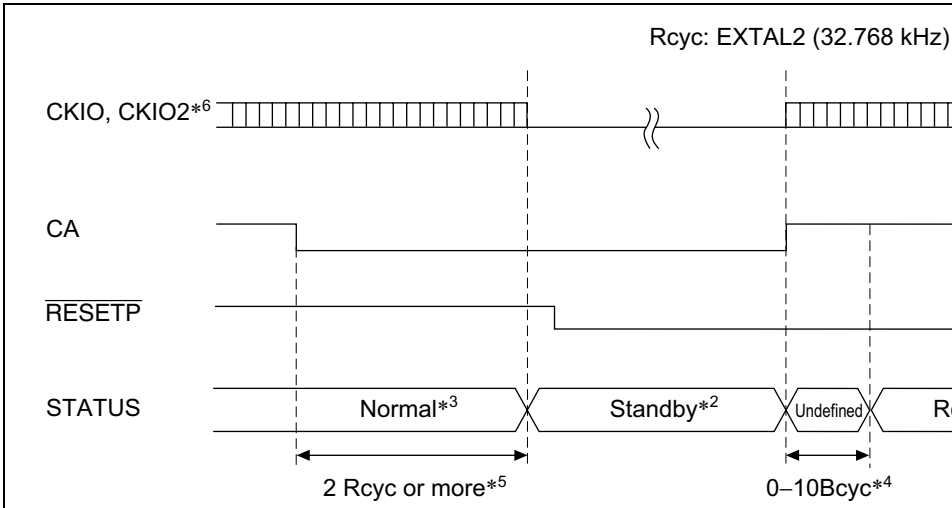
Hold the CA pin low in hardware standby mode.

8.7.2 Canceling Hardware Standby Mode

Hardware standby mode can only be canceled by a power-on reset.

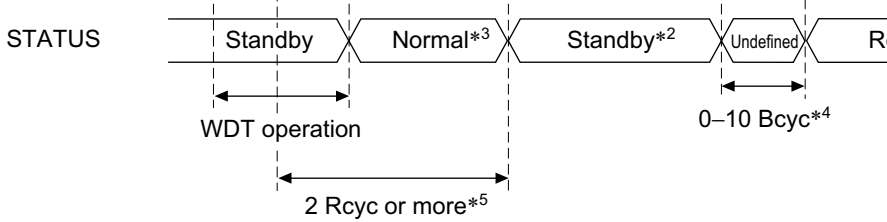
When the CA pin is driven high while the $\overline{\text{RESETP}}$ pin is low, clock oscillation is star the $\overline{\text{RESETP}}$ pin low until clock oscillation stabilizes. When the $\overline{\text{RESETP}}$ pin is drive CPU begins power-on reset processing.

Operation is not guaranteed in the event of an interrupt or manual reset.



Notes: 1. Reset: HH (STATUS1 high, STATUS0 high)
2. Standby: LH (STATUS1 low, STATUS0 high)
3. Normal: LL (STATUS1 low, STATUS0 low)
4. Bcyc: Bus clock cycle
5. Rcyc: EXTAL2 (32.768 kHz) cycle
6. The CKIO2 output is available only in clock modes 0, 1, and 2.

Figure 8.10 Hardware Standby Mode (When CA Goes Low in Normal Operation)



Notes: 1. Reset: HH (STATUS1 high, STATUS0 high)
2. Standby: LH (STATUS1 low, STATUS0 high)
3. Normal: LL (STATUS1 low, STATUS0 low)
4. Bcyc: Bus clock cycle
5. Rcyc: EXTAL2 (32.768 kHz) cycle
6. The CKIO2 output is available only in clock modes 0, 1, and 2.

Figure 8.11 Hardware Standby Mode Timing
(When CA Goes Low during WDT Operation on Standby Mode Cancella

9.1.1 Features

The CPG has the following features:

- Four clock modes: Selection of four clock modes for different frequency ranges, p consumption, direct crystal input, and external clock input.
- Three clocks generated independently: An internal clock for the CPU, cache, and T peripheral clock (Pϕ) for the on-chip peripheral modules; and a bus clock (CKIO) external bus interface.
- Frequency change function: Internal and peripheral clock frequencies can be chang independently using the PLL circuit and divider circuit within the CPG. Frequenci changed by software using frequency control register (FRQCR) settings.
- Power-down mode control: The clock can be stopped for sleep mode and standby specific modules can be stopped using the module standby function.

The WDT has the following features:

- Can be used to ensure the clock settling time: Use the WDT to cancel standby mod temporary standbys which occur when the clock frequency is changed.
- Can switch between watchdog timer mode and interval timer mode.
- Generates internal resets in watchdog timer mode: Internal resets occur after count Selection of power-on reset or manual reset.
- Generates interrupts in interval timer mode: Internal timer interrupts occur after co overflow.
- Selection of eight counter input clocks. Eight clocks (×1 to ×1/4096) can be obtain dividing the peripheral clock.

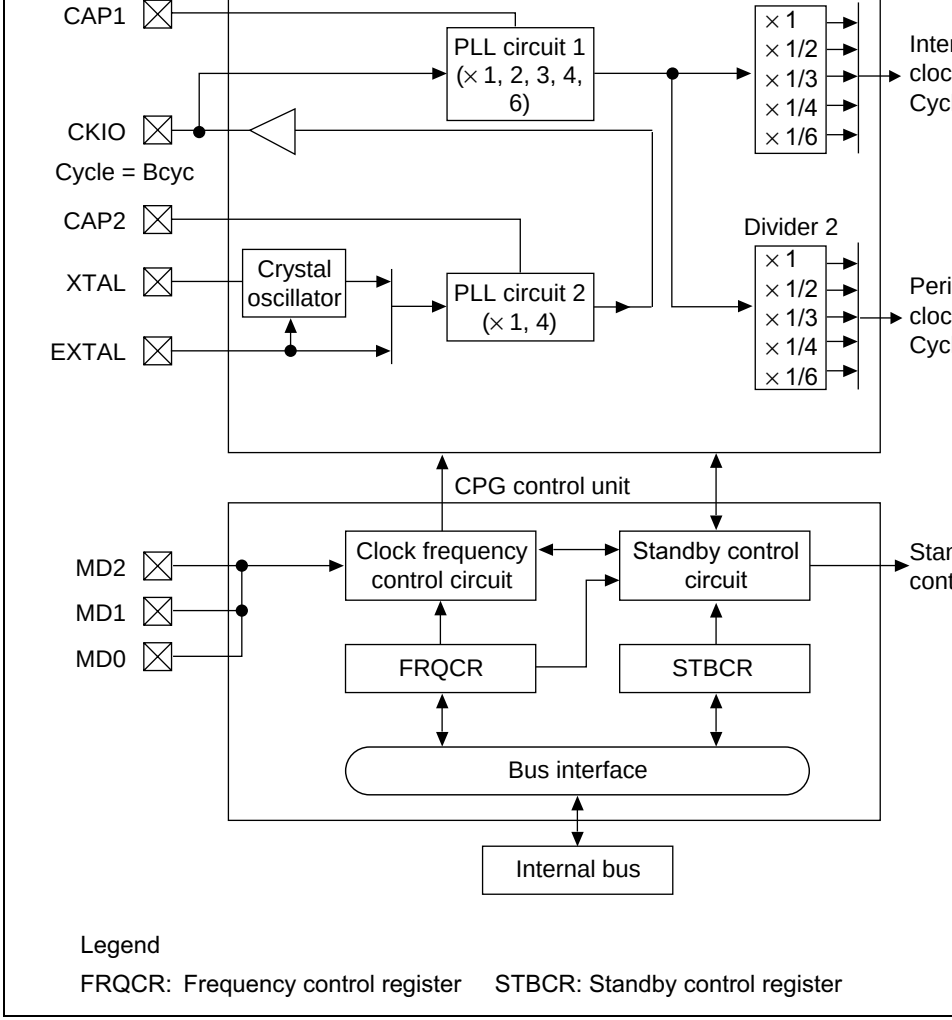


Figure 9.1 Block Diagram of Clock Pulse Generator

See table 9.3 for more information on clock operation modes.

3. Crystal Oscillator: This oscillator is used when a crystal oscillator element is conn XTAL and EXTAL pins. It operates according to the clock operating mode setting
4. Divider 1: Divider 1 generates a clock at the operating frequency used by the inter The operating frequency can be 1, 1/2, 1/3, 1/4 or 1/6 times the output frequency o circuit 1, as long as it is not lower than the CKIO pin clock frequency. The divisio in the frequency control register.
5. Divider 2: Divider 2 generates a clock at the operating frequency used by the perip The operating frequency can be 1, 1/2, 1/3, 1/4 or 1/6 times the output frequency o circuit 1 or the CKIO pin clock frequency, as long as it is not higher than the CKI frequency. The division ratio is set in the frequency control register.
6. Clock Frequency Control Circuit: The clock frequency control circuit controls the frequency using the MD pins and the frequency control register.
7. Standby Control Circuit: The standby control circuit controls the state of the clock generator and other modules during clock switching and sleep/standby modes.
8. Frequency Control Register: The frequency control register has control bits assign following functions: the frequency multiplication ratio of PLL 1, and the frequenc ratio of the internal clock and the peripheral clock.
9. Standby Control Register: The standby control register has bits for controlling the modes. See section 8, Power-Down Modes, for more information.

	MD2	I	
Crystal I/O pins (clock input pins)	XTAL	O	Connects a crystal oscillator
	EXTAL	I	Connects a crystal oscillator. Also used to inpu external clock
Clock I/O pin	CKIO	I/O	Inputs or outputs an external clock
Capacitor connection pins for PLL	CAP1	I	Connects capacitor for PLL circuit 1 operation (recommended value 470 pF)
	CAP2	I	Connects capacitor for PLL circuit 2 operation (recommended value 470 pF)

9.2.3 CPG Register Configuration

Table 9.2 shows the CPG register configuration.

Table 9.2 CPG Register

Register Name	Abbreviation	R/W	Initial Value	Address	Ac
Frequency control register	FRQCR	R/W	H'0102	H'FFFFFF80	16

0	0	0	0	EXTAL	CKIO	On, multi-plication ratio: 1	On	PLL1 output	PLL1
1	0	0	1	EXTAL	CKIO	On, multi-plication ratio: 4	On	PLL1 output	PLL1
2	0	1	0	Crystal oscillator	CKIO	On, multi-plication ratio: 4	On	PLL1 output	PLL1
7	1	1	1	CKIO	—	Off	On	PLL1 output	PLL1
—	Except above value			Reserved					

Mode 0: An external clock is input from the EXTAL pin and undergoes waveform sh
PLL circuit 2 before being supplied inside the chip. PLL circuit 1 is constantly on. An
frequency of 25 MHz to 66.67 MHz can be used, and the CKIO frequency range is 25
66.67 MHz.

Mode 1: An external clock is input from the EXTAL pin and its frequency is multipli
PLL circuit 2 before being supplied inside the chip, allowing a low-frequency externa
used. An input clock frequency of 6.25 MHz to 16.67 MHz can be used, and the CKI
range is 25 MHz to 66.67 MHz.

Mode 2: The on-chip crystal oscillator operates, with the oscillation frequency being
by 4 by PLL circuit 2 before being supplied inside the chip, allowing a low crystal fre
used. A crystal oscillation frequency of 6.25 MHz to 16.67 MHz can be used, and the
frequency range is 25 MHz to 66.67 MHz.

As PLL circuit 1 compensates for fluctuations in the CKIO pin load, this mode is suitab
connection of synchronous DRAM.

Table 9.4 Available Combinations of Clock Mode and FRQCR Values

Clock Mode	FRQCR	PLL1	PLL2	Clock Rate* (I:B:P)	Input Frequency Range	CKIO Fre Range
0	H'0100	ON (× 1)	ON (× 1)	1:1:1	25 MHz to 33.34 MHz	25 MHz to
	H'0101	ON (× 1)	ON (× 1)	1:1:1/2	25 MHz to 66.67 MHz	25 MHz to
	H'0102	ON (× 1)	ON (× 1)	1:1:1/4	25 MHz to 66.67 MHz	25 MHz to
	H'0111	ON (× 2)	ON (× 1)	2:1:1	25 MHz to 33.34 MHz	25 MHz to
	H'0112	ON (× 2)	ON (× 1)	2:1:1/2	25 MHz to 66.67 MHz	25 MHz to
	H'0115	ON (× 2)	ON (× 1)	1:1:1	25 MHz to 33.34 MHz	25 MHz to
	H'0116	ON (× 2)	ON (× 1)	1:1:1/2	25 MHz to 66.67 MHz	25 MHz to
	H'0122	ON (× 4)	ON (× 1)	4:1:1	25 MHz to 33.34 MHz	25 MHz to
	H'0126	ON (× 4)	ON (× 1)	2:1:1	25 MHz to 33.34 MHz	25 MHz to
	H'012A	ON (× 4)	ON (× 1)	1:1:1	25 MHz to 33.34 MHz	25 MHz to
	H'A100	ON (× 3)	ON (× 1)	3:1:1	25 MHz to 33.34 MHz	25 MHz to
	H'A101	ON (× 3)	ON (× 1)	3:1:1/2	25 MHz to 66.67 MHz	25 MHz to
	H'E100	ON (× 3)	ON (× 1)	1:1:1	25 MHz to 33.34 MHz	25 MHz to
	H'E101	ON (× 3)	ON (× 1)	1:1:1/2	25 MHz to 66.67 MHz	25 MHz to
	H'A111	ON (× 6)	ON (× 1)	6:1:1	25 MHz to 33.34 MHz	25 MHz to

	H'0116	ON (× 2)	ON (× 4)	4:4:2	6.25 MHz to 16.67 MHz	25 MHz t
	H'0122	ON (× 4)	ON (× 4)	16:4:4	6.25 MHz to 8.34 MHz	25 MHz t
	H'0126	ON (× 4)	ON (× 4)	8:4:4	6.25 MHz to 8.34 MHz	25 MHz t
	H'012A	ON (× 4)	ON (× 4)	4:4:4	6.25 MHz to 8.34 MHz	25 MHz t
	H'A100	ON (× 3)	ON (× 4)	12:4:4	6.25 MHz to 8.34 MHz	25 MHz t
	H'A101	ON (× 3)	ON (× 4)	12:4:2	6.25 MHz to 16.67 MHz	25 MHz t
	H'E100	ON (× 3)	ON (× 4)	4:4:4	6.25 MHz to 8.34 MHz	25 MHz t
	H'E101	ON (× 3)	ON (× 4)	4:4:2	6.25 MHz to 16.67 MHz	25 MHz t
	H'A111	ON (× 6)	ON (× 4)	24:4:4	6.25 MHz to 8.34 MHz	25 MHz t
7	H'0100	ON (× 1)	OFF	1:1:1	25 MHz to 33.34 MHz	25 MHz t
	H'0101	ON (× 1)	OFF	1:1:1/2	25 MHz to 66.67 MHz	25 MHz t
	H'0102	ON (× 1)	OFF	1:1:1/4	25 MHz to 66.67 MHz	25 MHz t
	H'0111	ON (× 2)	OFF	2:1:1	25 MHz to 33.34 MHz	25 MHz t
	H'0112	ON (× 2)	OFF	2:1:1/2	25 MHz to 66.67 MHz	25 MHz t
	H'0115	ON (× 2)	OFF	1:1:1	25 MHz to 33.34 MHz	25 MHz t
	H'0116	ON (× 2)	OFF	1:1:1/2	25 MHz to 66.67 MHz	25 MHz t
	H'0122	ON (× 4)	OFF	4:1:1	25 MHz to 33.34 MHz	25 MHz t
	H'0126	ON (× 4)	OFF	2:1:1	25 MHz to 33.34 MHz	25 MHz t
	H'012A	ON (× 4)	OFF	1:1:1	25 MHz to 33.34 MHz	25 MHz t
	H'A100	ON (× 3)	OFF	3:1:1	25 MHz to 33.34 MHz	25 MHz t
	H'A101	ON (× 3)	OFF	3:1:1/2	25 MHz to 66.67 MHz	25 MHz t
	H'E100	ON (× 3)	OFF	1:1:1	25 MHz to 33.34 MHz	25 MHz t
	H'E101	ON (× 3)	OFF	1:1:1/2	25 MHz to 66.67 MHz	25 MHz t
	H'A111	ON (× 6)	OFF	6:1:1	25 MHz to 33.34 MHz	25 MHz t

Do not set values other than those in the table above in the FRQCR register.

Note: * Taking input clock as 1.

- The peripheral clock frequency should not be set higher than the frequency of th pin, higher than 33.34 MHz.

3. The output frequency of PLL circuit 1 is the product of the CKIO frequency and the multiplication ratio of PLL circuit 1.
4. × 1, × 2, × 3, × 4, or × 6 can be used as the multiplication ratio of PLL circuit 1. × 1 × 1/3, × 1/4, and × 1/6 can be selected as the division ratios of dividers 1 and 2. Set the frequency control register. The on/off state of PLL circuit 2 is determined by the

FRQCR is initialized to H'0102 by a power-on reset, but retains its value in a manual r
standby mode.

FRQCR:

Bit:	15	14	13	12	11	10	9
	STC2	IFC2	PFC2	—	—	—	—
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R	R	R	R

Bit:	7	6	5	4	3	2	1
	—	—	STC1	STC0	IFC1	IFC0	PFC
Initial value:	0	0	0	0	0	0	1
R/W:	R	R	R/W	R/W	R/W	R/W	R/W

Bits 15, 5, and 4—Frequency Multiplication Ratio (STC): These bits specify the fr
multiplication ratio of PLL circuit 1.

Bit 15: STC2	Bit 5: STC1	Bit 4: STC0	Description
0	0	0	× 1
0	0	1	× 2
1	0	0	× 3
0	1	0	× 4
1	0	1	× 6
Except above value			Reserved

Except above value	Reserved (Setting prohibited)

Note: Do not set the internal clock frequency lower than the CKIO pin frequency.

Bits 13, 1, and 0—Peripheral Clock Frequency Division Ratio (PFC): These bits sp division ratio of the peripheral clock frequency with respect to the frequency of the outp frequency of PLL circuit 1 or the frequency of the CKIO pin.

Bit 13: PFC2	Bit 1: PFC1	Bit 0: PFC0	Description
0	0	0	× 1
0	0	1	× 1/2
1	0	0	× 1/3
0	1	0	× 1/4 (Initial value)
1	0	1	× 1/6
Except above value			Reserved (Setting prohibited)

Note: Do not set the peripheral clock frequency higher than the CKIO pin frequency.

Bits 12 to 9, 7, and 6—Reserved: These bits are always read as 0. The write value sho be 0.

Bit 8—Reserved: This bit is always read as 1. The write value should always be 1.

A PLL settling time is required when the multiplication rate of PLL circuit 1 is change
chip WDT counts the settling time.

1. In the initial state, the multiplication rate of PLL circuit 1 is 1.
2. Set a value that will become the specified oscillation settling time in the WDT and
   WDT. The following must be set:
   WTCSR register TME bit = 0: WDT stops
   WTCSR register CKS2–CKS0 bits: Division ratio of WDT count clock
   WTCNT counter: Initial counter value
3. Set the desired value in the STC2 to STC0 bits. The division ratio can also be set i
   IFC0 bits and PFC2–PFC0 bits.
4. The processor pauses internally and the WDT starts incrementing. In clock modes
   the internal and peripheral clocks both stop. (except for the peripheral clock suppli
   WDT)
5. Supply of the clock that has been set begins at WDT count overflow, and the proc
   operating again. The WDT stops after it overflows.

When the following three conditions are all met, FRQCR should not be changed while
transfer is in progress.

- Bits IFC2 to IFC0 are changed.
- STC2 to STC0 are not changed.
- The clock ratio of Iϕ (on-chip clock) to Bϕ (bus clock) after the change is other th

9.5.2 Changing the Division Ratio

The WDT will not count unless the multiplication ratio is changed simultaneously.

1. In the initial state, IFC2–IFC0 = 000 and PFC2–PFC0 = 010.
2. Set the IFC2, IFC1, IFC0, PFC2, PFC1, and PFC0 bits to the new division ratio. T
   that can be set are limited by the clock mode and the multiplication ratio of PLL ci
   that if the wrong value is set, the processor will malfunction.
3. The clock is immediately supplied at the new division ratio.

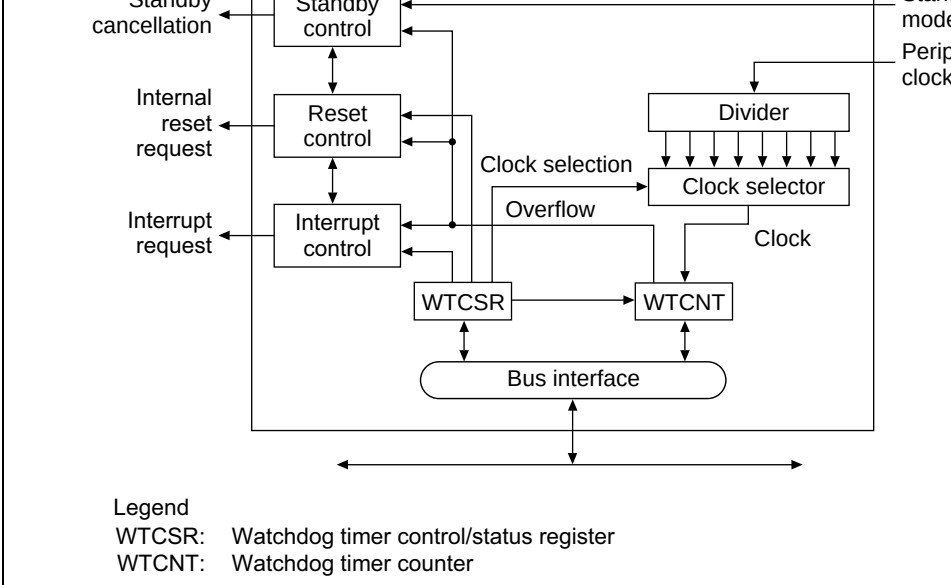


Figure 9.2 Block Diagram of WDT

9.6.2 Register Configuration

The WDT has two registers that select the clock, switch the timer mode, and perform o functions. Table 9.5 shows the WDT registers.

Table 9.5 Register Configuration

Name	Abbreviation	R/W	Initial Value	Address	Ac
Watchdog timer counter	WTCNT	R/W*	H'00	H'FFFFFF84	R: W:
Watchdog timer control/status register	WTCSR	R/W*	H'00	H'FFFFFF86	R: W:

Note: * Write with word access. Write with H'5A and H'A5, respectively, in the upper by longword writes are not possible. Read with byte access.

word access to write to the WTCNT counter, with H'5A in the upper byte. Use byte ac
WTCNT.

Bit:	7	6	5	4	3	2	1
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

9.7.2 Watchdog Timer Control/Status Register (WTCSR)

The watchdog timer control/status register (WTCSR) is an 8-bit readable/writable regi
composed of bits to select the clock used for the count, bits to select the timer mode, a
flags. WTCSR differs from other registers in that it is more difficult to write to. See se
Notes on Register Access, for details. Its address is H'FFFFFF86. The WTCSR registe
initialized to H'00 only by a power-on reset through the $\overline{\text{RESETP}}$ pin. When a WDT o
causes an internal reset, WTCSR retains its value. When used to count the clock settli
canceling a standby, it retains its value after counter overflow. Use word access to wri
WTCSR counter, with H'A5 in the upper byte. Use byte access to read WTCSR.

Bit:	7	6	5	4	3	2	1
	TME	WT/$\overline{\text{IT}}$	RSTS	WOVF	IOVF	CKS2	CKS
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7—Timer Enable (TME): Starts and stops timer operation. Clear this bit to 0 wh
WDT in standby mode or when changing the clock frequency.

Bit 7: TME	Description
0	Timer disabled: Count-up stops and WTCNT value is retained
1	Timer enabled

Bit 5—Reset Select (RSTS): Selects the type of reset when WTCNT overflows in wat
timer mode. In interval timer mode, this setting is ignored.

Bit 5: RSTS	Description	
0	Power-on reset	(I
1	Manual reset	

Note: $\overline{\text{RESETOUT}}$ is output.

Bit 4—Watchdog Timer Overflow (WOVF): Indicates that the WTCNT has overflov
watchdog timer mode. This bit is not set in interval timer mode.

Bit 4: WOVF	Description	
0	No overflow	(I
1	WTCNT has overflowed in watchdog timer mode	

Bit 3—Interval Timer Overflow (IOVF): Indicates that WTCNT has overflowed in i
timer mode. This bit is not set in watchdog timer mode.

Bit 3: IOVF	Description	
0	No overflow	(I
1	WTCNT has overflowed in interval timer mode	

Bits 2 to 0—Clock Select 2 to 0 (CKS2 to CKS0): These bits select the clock to be us
WTCNT count from the eight types obtainable by dividing the peripheral clock. The ov
period in the table is the value when the peripheral clock (Pϕ) is 15 MHz.

1	0	1/1024	17.48 ms
	1	1/4096	69.91 ms

Note: If bits CKS2–CKS0 are modified when the WDT is running, the up-count may n performed correctly. Ensure that these bits are modified only when the WDT is

9.7.3 Notes on Register Access

The watchdog timer counter (WTCNT) and watchdog timer control/status register (W more difficult to write to than other registers. The procedure for writing to these regist below.

Writing to WTCNT and WTCSR: These registers must be written to using a word t instruction. They cannot be written to with a byte or longword transfer instruction. Wh to WTCNT, set the upper byte to H'5A and transfer the lower byte as the write data, as figure 9.3. When writing to WTCSR, set the upper byte to H'A5 and transfer the lowe write data. This transfer procedure writes the lower byte data to WTCNT or WTCSR.

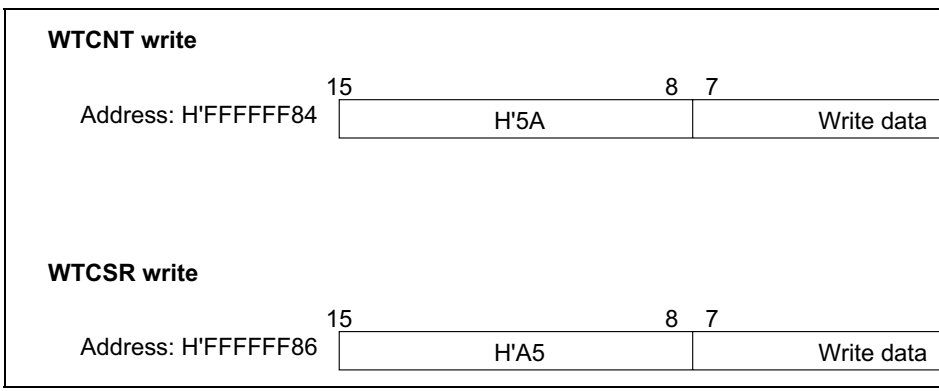


Figure 9.3 Writing to WTCNT and WTCSR

TME bit is 1, an erroneous reset or interval timer interrupt may be generated when t overflows.

2. Set the type of count clock used in the CKS2–CKS0 bits in WTCSR and the initial v the counter in the WTCNT counter. These values should ensure that the time till cou overflow is longer than the clock oscillation settling time.
3. Switch to standby mode by executing a SLEEP instruction to stop the clock.
4. The WDT starts counting by detecting the edge change of the NMI signal or detecti interrupts.
5. When the WDT count overflows, the CPG starts supplying the clock and the proces resumes operation. The WOVF flag in WTCSR is not set when this happens.
6. Since the WDT continues counting from H'00, set the STBY bit in the STBCR regis the interrupt handling routine and this will stop the WDT. When the STBY bit rema the SH7709S again enters standby mode when the WDT has counted up to H'80. Th mode can be canceled by a power-on reset.

9.8.2 Changing the Frequency

To change the frequency used by the PLL, use the WDT. When changing the frequency switching the divider, do not use the WDT.

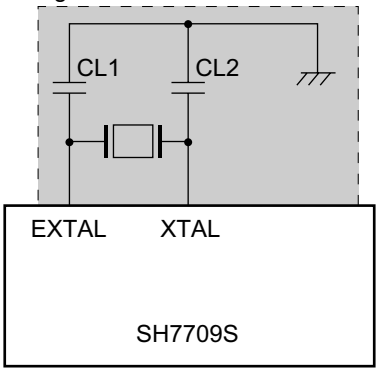
1. Before changing the frequency, always clear the TME bit in WTCSR to 0. When th is 1, an erroneous reset or interval timer interrupt may be generated when the count
2. Set the type of count clock used in the CKS2–CKS0 bits of WTCSR and the initial the counter in the WTCNT counter. These values should ensure that the time till cou overflow is longer than the clock oscillation settling time.
3. When the frequency control register (FRQCR) is written to, the clock stops and the enters standby mode temporarily. The WDT starts counting.
4. When the WDT count overflows, the CPG resumes supplying the clock and the proc resumes operation. The WOVF flag in WTCSR is not set when this happens.
5. The counter stops at a value of H'00 or H'01. The stop value depends on the clock ra

1. Set the WT/$\overline{IT}$ bit in the WTCSR register to 1, set the reset type in the RSTS bit, s count clock in the CKS2–CKS0 bits, and set the initial value of the counter in the V counter.
2. Set the TME bit in WTCSR to 1 to start the count in watchdog timer mode.
3. While operating in watchdog timer mode, rewrite the counter periodically to H'00 the counter from overflowing.
4. When the counter overflows, the WDT sets the WOVF flag in WTCSR to 1 and ge type of reset specified by the RSTS bit. The counter then resumes counting. When generated, a low level is output at the $\overline{RESETOUT}$ pin, and a high level at the STA STATUS1 pins. The output period is approximately 1 count clock cycle in the cas on reset, and approximately 5 peripheral clock cycles in the case of a manual reset

9.8.4 Using Interval Timer Mode

When operating in interval timer mode, interval timer interrupts are generated at every the counter. This enables interrupts to be generated at set periods.

1. Clear the WT/$\overline{IT}$ bit in the WTCSR register to 0, set the type of count clock in the CKS0 bits, and set the initial value of the counter in the WTCNT counter.
2. Set the TME bit in WTCSR to 1 to start the count in interval timer mode.
3. When the counter overflows, the WDT sets the IOVF flag in WTCSR to 1 and an i timer interrupt request is sent to the INTC. The counter then resumes counting.



Note: The values for CL1 and CL2 should be determined after consultation with the crystal manufacturer.

Figure 9.4 Points for Attention when Using Crystal Resonator

Decoupling Capacitors: Insert a laminated ceramic capacitor of 0.1 to 1 µF as a passiv for each V_{SS}/V_{CC} pair. Mount the passive capacitors close to the SH7709S power supply use components with a frequency characteristic suitable for the chip's operating frequer as a suitable capacitance value.

Digital system V_{SS}/V_{CC} pairs: 19-21, 27-29, 33-35, 45-47, 57-59, 69-71, 79-81, 83-85, 9 111, 132-134, 153-154, 161-163, 173-175, 181-183, 205-208

On-chip oscillator V_{SS}/V_{CC} pairs: 3-6, 145-147, 148-150

Note: The pin numbers above apply to LQFP and HQFP packages.

When Using a PLL Oscillator Circuit: Keep the wiring from the PLL V_{CC} and V_{SS} co pattern to the power supply pins short, and make the pattern width large, to minimize th inductance component. Ground the oscillation stabilization capacitors C1 and C2 to V_{S} and V_{SS} (PLL2), respectively. Place C1 and C2 close to the CAP1 and CAP2 pins and c locate a wiring pattern in the vicinity. In clock mode 7, connect the EXTAL pin to V_{CC} leave the XTAL pin open.

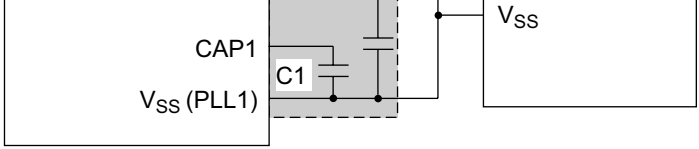


Figure 9.5 Points for Attention when Using PLL Oscillator Circuit

design and allowing high-speed data transfers in a compact system.

10.1.1 Features

The BSC has the following features:

- Physical address space is divided into six areas
  - A maximum 64 Mbytes for each of the six areas, 0, 2–6
  - Area bus width can be selected by register (area 0 is set by external pin)
  - Wait states can be inserted using the $\overline{\text{WAIT}}$ pin
  - Wait state insertion can be controlled through software. Register settings can b specify the insertion of 1–10 cycles independently for each area (1–38 cycles f and 6 and the PCMCIA interface only)
  - The type of memory connected can be specified for each area, and control sign output for direct memory connection
  - Wait cycles are automatically inserted to avoid data bus conflict for continuous accesses to different areas or writes directly following reads in the same area
- Direct interface to synchronous DRAM
  - Multiplexes row/column addresses according to synchronous DRAM capacity
  - Supports burst operation
  - Supports bank active mode
  - Has both auto-refresh and self-refresh functions
  - Controls timing of synchronous DRAM direct-connection control signals acco register setting
- Burst ROM interface
  - Insertion of wait states controllable through software
  - Register setting control of burst transfers
- PCMCIA direct-connection interface
  - Insertion of wait states controllable through software
  - Bus sizing function for I/O bus width (only in little-endian mode)

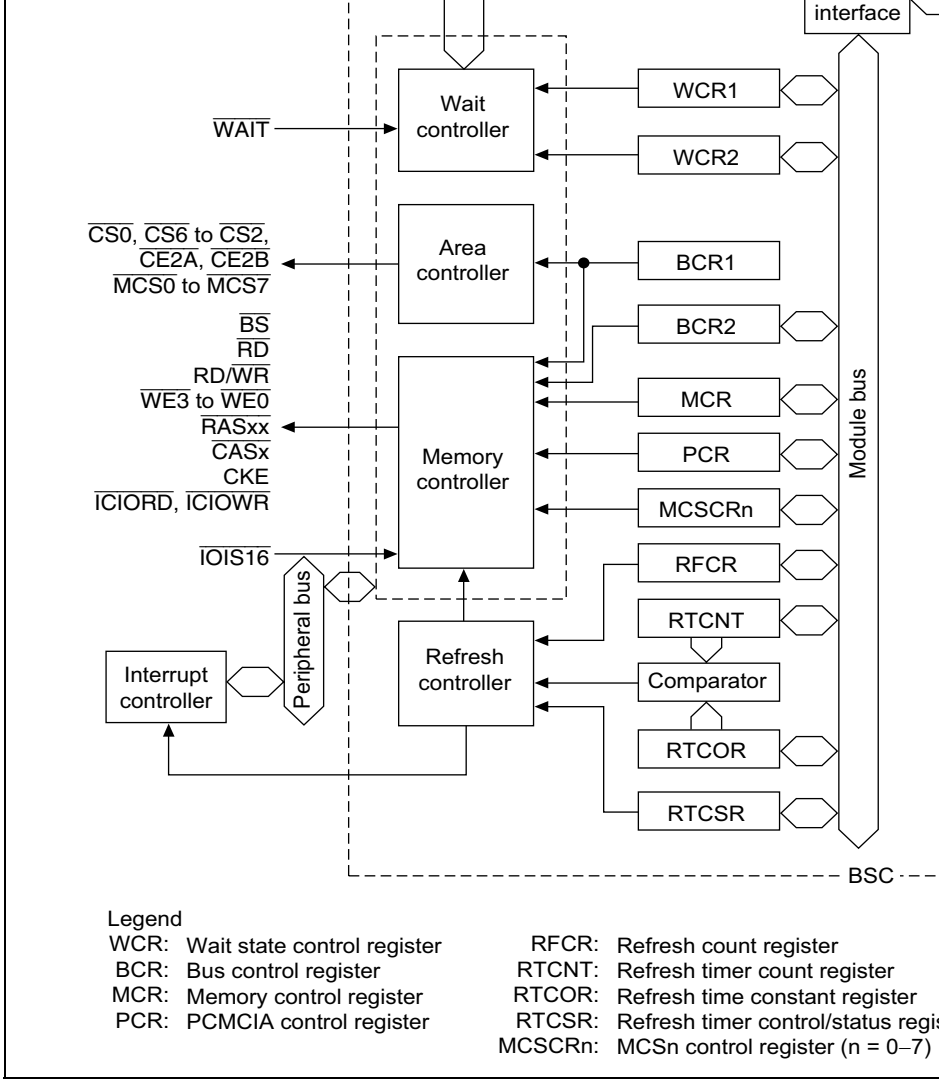


Figure 10.1 Block Diagram of Bus State Controller

	D31–D16	I/O	Data I/O when using 32-bit bus width
Bus cycle start	$\overline{BS}$	O	Shows start of bus cycle. During burst tra asserted every data cycle.
Chip select 0, 2–4	$\overline{CS0}$, $\overline{CS2}$–$\overline{CS4}$	O	Chip select signals to indicate area being
Chip select 5, 6	$\overline{CS5}$/$\overline{CE1A}$, $\overline{CS6}$/$\overline{CE1B}$	O	Chip select signals to indicate area being $\overline{CS5}$/$\overline{CE1A}$ and $\overline{CS6}$/$\overline{CE1B}$ can also be u $\overline{CE1A}$ and $\overline{CE1B}$ of PCMCIA.
PCMCIA card select	$\overline{CE2A}$, $\overline{CE2B}$	O	$\overline{CE2A}$ and $\overline{CE2B}$ signals when PCMCIA
Read/write	RD/$\overline{WR}$	O	Data bus direction indication signal. PCM indication signal.
Row address strobe 3L	$\overline{RAS3L}$	O	When synchronous DRAM is used, $\overline{RAS}$ 32-Mbyte address and 64-Mbyte addres
Row address strobe 3U	$\overline{RAS3U}$	O	When synchronous DRAM is used, $\overline{RAS}$ upper 32-Mbyte address.
Column address strobe	$\overline{CASL}$	O	When synchronous DRAM is used, $\overline{CAS}$ lower 32-Mbyte address and 64-Mbyte a
Column address strobe LH	$\overline{CASU}$	O	When synchronous DRAM is used, $\overline{CAS}$ upper 32-Mbyte address.
Data enable 0	$\overline{WE0}$/DQMLL	O	When memory other than synchronous D used, D7–D0 write strobe signal. When synchronous DRAM is used, selects D7-
Data enable 1	$\overline{WE1}$/DQMLU/ $\overline{WE}$	O	When memory other than synchronous D PCMCIA is used, D15–D8 write strobe s synchronous DRAM is used, selects D15 PCMCIA is used, strobe signal indicating cycle.
Data enable 2	$\overline{WE2}$/DQMUL/ $\overline{ICIORD}$	O	When memory other than synchronous D PCMCIA is used, D23–D16 write strobe When synchronous DRAM is used, selec D16. When PCMCIA is used, strobe sign indicating I/O read.

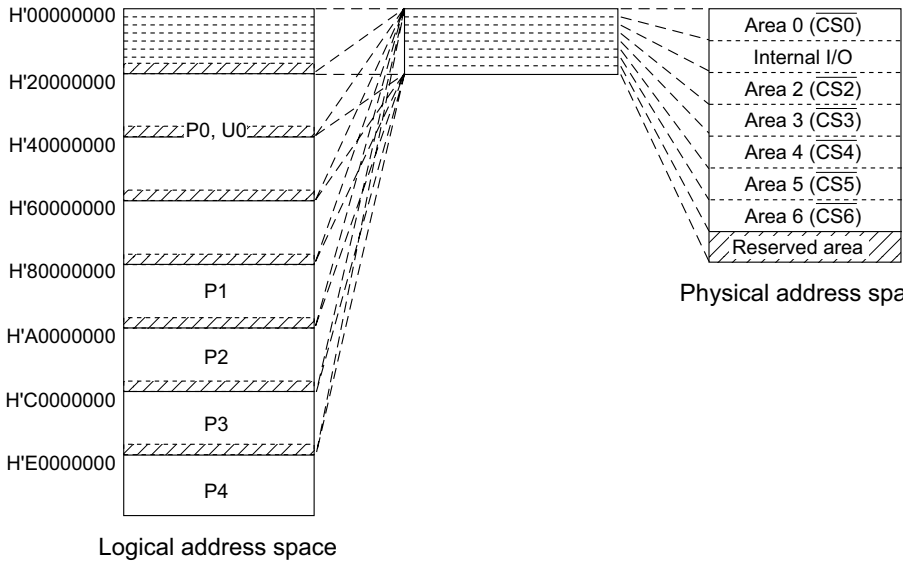
IOIS16	$\overline{\text{IOIS16}}$	I	Signal indicating PCMCIA 16-bit I/O. V little-endian mode.
Bus release request	$\overline{\text{BREQ}}$	I	Bus release request signal
Bus release acknowledgment	$\overline{\text{BACK}}$	O	Bus release acknowledge signal
Mask ROM chip select	$\overline{\text{MCS[0]}}$– $\overline{\text{MCS[7]}}$	O	Chip select signal for mask ROM conne 0 or 2.

Bus control register 1	BCR1	R/W	H'0000	H'FFFFFF60	1
Bus control register 2	BCR2	R/W	H'3FF0	H'FFFFFF62	1
Wait state control register 1	WCR1	R/W	H'3FF3	H'FFFFFF64	1
Wait state control register 2	WCR2	R/W	H'FFFF	H'FFFFFF66	1
Individual memory control register	MCR	R/W	H'0000	H'FFFFFF68	1
PCMCIA control register	PCR	R/W	H'0000	H'FFFFFF6C	1
Refresh timer control/status register	RTCSR	R/W	H'0000	H'FFFFFF6E	1
Refresh timer counter	RTCNT	R/W	H'0000	H'FFFFFF70	1
Refresh time constant register	RTCOR	R/W	H'0000	H'FFFFFF72	1
Refresh count register	RFCR	R/W	H'0000	H'FFFFFF74	1
Synchronous DRAM mode register, area 2	SDMR	W	—	H'FFFFD000–H'FFFFDFFF	8
Synchronous DRAM mode register, area 3				H'FFFFE000–H'FFFFEFFF	
MCS0 control register	MCSCR0	R/W	H'0000	H'FFFFFF50	1
MCS1 control register	MCSCR1	R/W	H'0000	H'FFFFFF52	1
MCS2 control register	MCSCR2	R/W	H'0000	H'FFFFFF54	1
MCS3 control register	MCSCR3	R/W	H'0000	H'FFFFFF56	1
MCS4 control register	MCSCR4	R/W	H'0000	H'FFFFFF58	1
MCS5 control register	MCSCR5	R/W	H'0000	H'FFFFFF5A	1
MCS6 control register	MCSCR6	R/W	H'0000	H'FFFFFF5C	1
MCS7 control register	MCSCR7	R/W	H'0000	H'FFFFFF5E	1

Notes: For details, see section 10.2.7, Synchronous DRAM Mode Register (SDMR).

* Initialized by a power-on reset.

As shown in table 10.3, the SH7709S can be connected directly to six memory/PCMC areas, and it outputs chip select signals ($\overline{CS0}$, $\overline{CS2}$–$\overline{CS6}$, $\overline{CE2A}$, $\overline{CE2B}$) for each of the asserted during area 0 access; $\overline{CS6}$ is asserted during area 6 access. When PCMCIA in selected in area 5 or 6, in addition to $\overline{CS5}$/$\overline{CS6}$, $\overline{CE2A}$/$\overline{CE2B}$ are asserted for the corre bytes accessed.



Note: For logical address spaces P0 and P3, when the memory management unit (M on, it can optionally generate a physical address for the logical address. This can be applied when the MMU is off, and when the MMU is on and each phys address corresponding to a logical address is equal except for the upper three When translating logical addresses to arbitrary physical addresses, refer to ta

Figure 10.2 Correspondence between Logical Address Space and Physical Add

2	Ordinary memory*1, synchronous DRAM	H'08000000 to H'0BFFFFFF	64 Mbytes	8,
		H'08000000 + H'20000000 × n to H'0BFFFFFF + H'20000000 × n	Shadow	n =
3	Ordinary memory*1, synchronous DRAM	H'0C000000 to H'0FFFFFFF	64 Mbytes	8,
		H'0C000000 + H'20000000 × n to H'0FFFFFFF + H'20000000 × n	Shadow	n =
4	Ordinary memory*1	H'10000000 to H'13FFFFFF	64 Mbytes	8,
		H'10000000 + H'20000000 × n to H'13FFFFFF + H'20000000 × n	Shadow	n =
5	Ordinary memory*1, PCMCIA, burst ROM Ordinary memory, burst ROM	H'14000000 to H'15FFFFFF	32 Mbytes	8,
		H'16000000 to H'17FFFFFF	32 Mbytes	
		H'14000000 + H'20000000 × n to H'17FFFFFF + H'20000000 × n	Shadow	n =
6	Ordinary memory*1, PCMCIA, burst ROM	H'18000000 to H'19FFFFFF	32 Mbytes	8,
		H'1A000000 to H'1BFFFFFF		
		H'18000000 + H'20000000 × n to H'1BFFFFFF + H'20000000 × n	Shadow	n =
7*6	Reserved area	H'1C000000 + H'20000000 × n to H'1FFFFFFF + H'20000000 × n		n =

Notes: 1. Memory with interface such as SRAM or ROM.
2. Use external pin to specify memory bus width.
3. Use register to specify memory bus width.
4. With synchronous DRAM interfaces, bus width must be 16 or 32 bits.
5. With PCMCIA interface, bus width must be 8 or 16 bits.
6. Do not access the reserved area. If the reserved area is accessed, correct o cannot be guaranteed.
7. When the control register in area 1 is not used for address translation by the the first three bits of the logical address to 101 for allocation to the P2 space

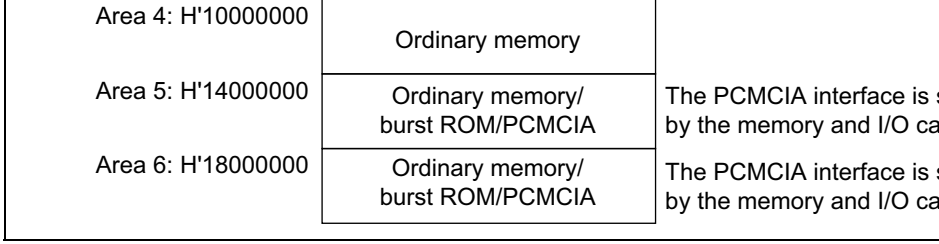


Figure 10.3 Physical Space Allocation

Memory Bus Width: The memory bus width in the SH7709S can be set for each area external pins can be used to select byte (8 bits), word (16 bits), or longword (32 bits) reset. The correspondence between the external pins (MD4 and MD3) and the memory shown in table below.

Table 10.4 Correspondence between External Pins (MD4 and MD3) and Memo

MD4	MD3	Memory Size
0	0	Reserved (Do not set)
0	1	8 bits
1	0	16 bits
1	1	32 bits

For areas 2–6, byte, word, and longword can be chosen for the bus width using bus co 2 (BCR2) whenever ordinary memory, ROM, or burst ROM are used. When the synch DRAM interface is used, word or longword can be chosen as the bus width.

When the PCMCIA interface is used, set the bus width to byte or word. When synchro DRAM is connected to both area 2 and area 3, set the same bus width for areas 2 and using the port function, set each of the bus widths to byte or word for all areas. For mo information, see section 10.2.2, Bus Control Register 2 (BCR2).

The SH7709S supports PCMCIA standard interface specifications in physical space are

The interfaces supported are basically the “IC memory card interface” and “I/O card int stipulated in JEIDA Specifications Ver. 4.2 (PCMCIA2.1).

Table 10.5 PCMCIA Interface Characteristics

Item	Feature
Access	Random access
Data bus	8/16 bits
Memory type	Mask ROM, OTPROM, EPROM, EEPROM, flash memory
Memory capacity	Maximum 32 Mbytes
I/O space capacity	Maximum 32 Mbytes
Other features	Dynamic bus sizing of I/O bus width* The PCMCIA interface can be accessed from the address area or non-address translation area.

Note: * Dynamic bus sizing of the I/O bus width is supported only in little-endian mode.

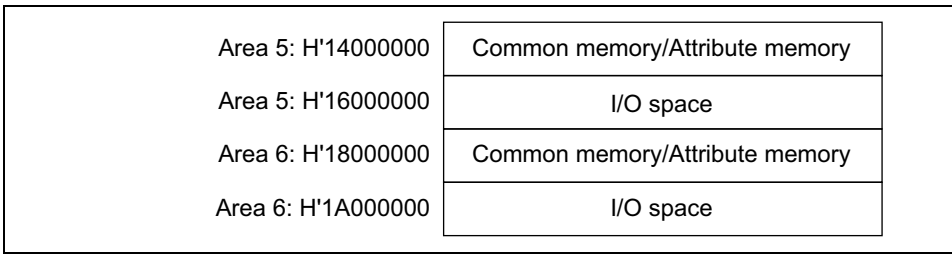


Figure 10.4 PCMCIA Space Allocation

5	D6	I/O	Data	D6	I/O	Data	D6
6	D7	I/O	Data	D7	I/O	Data	D7
7	$\overline{CE1}$	I	Card enable	$\overline{CE1}$	I	Card enable	$\overline{CE}$
8	A10	I	Address	A10	I	Address	A1
9	OE	I	Output enable	$\overline{OE}$	I	Output enable	$\overline{RD}$
10	A11	I	Address	A11	I	Address	A1
11	A9	I	Address	A9	I	Address	A9
12	A8	I	Address	A8	I	Address	A8
13	A13	I	Address	A13	I	Address	A1
14	A14	I	Address	A14	I	Address	A1
15	$\overline{WE}/\overline{PGM}$	I	Write enable	$\overline{WE}/\overline{PGM}$	I	Write enable	$\overline{W}$
16	RDY/$\overline{BSY}$	O	Ready/Busy	$\overline{IREQ}$	O	Ready/Busy	—
17	V_{CC}		Operation power	V_{CC}		Operation power	—
18	VPP1		Program power	VPP1		Program/ peripheral power	—
19	A16	I	Address	A16	I	Address	A1
20	A15	I	Address	A15	I	Address	A1
21	A12	I	Address	A12	I	Address	A1
22	A7	I	Address	A7	I	Address	A7
23	A6	I	Address	A6	I	Address	A6
24	A5	I	Address	A5	I	Address	A5
25	A4	I	Address	A4	I	Address	A4
26	A3	I	Address	A3	I	Address	A3
27	A2	I	Address	A2	I	Address	A2
28	A1	I	Address	A1	I	Address	A1
29	A0	I	Address	A0	I	Address	A0
30	D0	I/O	Data	D0	I/O	Data	D0

37	D11	I/O	Data	D11	I/O	Data	D11
38	D12	I/O	Data	D12	I/O	Data	D12
39	D13	I/O	Data	D13	I/O	Data	D13
40	D14	I/O	Data	D14	I/O	Data	D14
41	D15	I/O	Data	D15	I/O	Data	D15
42	$\overline{CE2}$	I	Card enable	$\overline{CE2}$	I	Card enable	$\overline{CE2}$
43	$\overline{VS1}$	I	Voltage sense 1	$\overline{VS1}$	I	Voltage sense 1	—
44	RFU		Reserved	$\overline{IORD}$	I	I/O read	$\overline{ICIO}$
45	RFU		Reserved	$\overline{IOWR}$	I	I/O write	$\overline{ICIO}$
46	A17	I	Address	A17	I	Address	A17
47	A18	I	Address	A18	I	Address	A18
48	A19	I	Address	A19	I	Address	A19
49	A20	I	Address	A20	I	Address	A20
50	A21	I	Address	A21	I	Address	A21
51	V_{CC}		Power supply	V_{CC}		Power supply	—
52	VPP2		Program power	VPP2		Program/ peripheral power	—
53	A22	I	Address	A22	I	Address	A22
54	A23	I	Address	A23	I	Address	A23
55	A24	I	Address	A24	I	Address	A24
56	A25	I	Address	A25	I	Address	A25
57	$\overline{VS2}$	I	Voltage sense 2	$\overline{VS2}$	I	Voltage sense 2	—
58	RESET	I	Reset	RESET	I	Reset	—
59	$\overline{WAIT}$	O	Wait request	$\overline{WAIT}$	O	Wait request	—
60	RFU		Reserved	$\overline{INPACK}$	O	Input acknowledge	—

65	D9	I/O	Data	D9	I/O	Data	D9
66	D10	I/O	Data	D10	I/O	Data	D1
67	$\overline{CD2}$	O	Card detection	$\overline{CD2}$	O	Card detection	—
68	GND		Ground	GND		Ground	—

10.2 BSC Registers

10.2.1 Bus Control Register 1 (BCR1)

Bus control register 1 (BCR1) is a 16-bit readable/writable register that sets the functi cycle state for each area. It is initialized to H'0000 by a power-on reset, but is not initi manual reset or in standby mode. Do not access external memory outside area 0 until register initialization is complete.

Bit:	15	14	13	12	11	10	9
	PULA	PULD	HIZMEM	HIZCNT	ENDIAN	A0BST1	A0BS
Initial value:	0	0	0	0	0/1*	0	0
R/W:	R/W	R/W	R/W	R/W	R	R/W	R/W

Bit:	7	6	5	4	3	2	1
	A5BST0	A6BST1	A6BST0	DRAM TP2	DRAM TP1	DRAM TP0	A5 PC
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: * Samples the value of the external pin (MD5) designating the endian in a powe

when not in use.

Bit 14: PULD	Description
0	Not pulled up (In
1	Pulled up

Bit 13—Hi-Z Memory Control (HIZMEM): Specifies the state of A25–A0, $\overline{BS}$, $\overline{CS}$, $\overline{WE}$/DQM, $\overline{RD}$, $\overline{CE2A}$, $\overline{CE2B}$ and DRAK0/1 in standby mode.

Bit 13: HIZMEM	Description
0	A25–A0, BS, CS, RD/WR, WE/DQM, RD, CE2A, CE2B and DRA Hi-Z in standby mode (In
1	A25–A0, BS, CS, RD/WR, WE/DQM, RD, CE2A, CE2B and DRA high in standby mode

Bit 12—High-Z Control (HIZCNT): Specifies the state of the $\overline{RAS}$ and $\overline{CAS}$ signals i
mode and when the bus is released.

Bit 12: HIZCNT	Description
0	$\overline{RAS}$ and $\overline{CAS}$ signals are high-impedance (High-Z) in standby m when bus is released (In
1	$\overline{RAS}$ and $\overline{CAS}$ signals are driven in standby mode and when bus

Bit 11—Endian Flag (ENDIAN): Samples the value of the external pin designating th
a power-on reset. The endian for all physical spaces is decided by this bit, which is read

Bit 11: ENDIAN	Description
0	(On reset) Endian setting external pin (MD5) is low. Indicates the is set as big-endian
1	(On reset) Endian setting external pin (MD5) is high. Indicates the is set as little-endian

1	0	Access area 0 accessed as burst ROM (8 c accesses). Can be used when bus width is Should not be specified when bus width is 1
	1	Access area 0 accessed as burst ROM (16 accesses). Can be used only when bus wid Should not be specified when bus width is 1

Bits 8 and 7—Area 5 Burst Enable (A5BST1, A5BST0): Specify whether to use bu
and PCMCIA burst mode in physical space area 5. When burst ROM and PCMCIA bu
are used, these bits set the number of burst transfers.

Bit 8: A5BST1	Bit 7: A5BST0	Description
0	0	Access area 5 accessed as ordinary memo (
	1	Burst access of area 5 (4 consecutive acce be used when bus width is 8, 16, or 32.
1	0	Burst access of area 5 (8 consecutive acce be used when bus width is 8 or 16. Should specified when bus width is 32.
	1	Burst access of area 5 (16 consecutive acc be used only when bus width is 8. Should n specified when bus width is 16 or 32.

Bits 6 and 5—Area 6 Burst Enable (A6BST1, A6BST0): Specify whether to use bu
and PCMCIA burst mode in physical space area 6. When burst ROM and PCMCIA bu
are used, these bits set the number of burst transfers.

be used only when bus width is 8. Should not
specified when bus width is 16 or 32.

Bits 4 to 2—Area 2, Area 3 Memory Type (DRAMTP2, DRAMTP1, DRAMTP0): the types of memory connected to physical space areas 2 and 3. Ordinary memory, such SRAM, or flash ROM, can be directly connected. Synchronous DRAM can also be dire connected.

Bit 4: DRAMTP2	Bit 3: DRAMTP1	Bit 2: DRAMTP0	Description
0	0	0	Areas 2 and 3 are ordinary mer (In
		1	Reserved (Setting prohibited)
	1	0	Area 2: ordinary memory; area synchronous DRAM*2
		1	Areas 2 and 3 are synchronous *2
1	0	0	Reserved (Setting prohibited)
		1	Reserved (Setting prohibited)
	1	0	Reserved (Setting prohibited)
		1	Reserved (Setting prohibited)

Notes: 1. When selecting this mode, set the same bus width for area 2 and area 3.
2. Do not access synchronous DRAM when clock ratio Iϕ:Bϕ = 1:1

Bit 1—Area 5 Bus Type (A5PCM): Designates whether to access physical space area PCMCIA space.

Bit 1: A5PCM	Description
0	Physical space area 5 accessed as ordinary memory (In
1	Physical space area 5 accessed as PCMCIA space

Bus control register 2 (BCR2) is a 16-bit readable/writable register that selects the bus
area and whether an 8-bit port is used or not. It is initialized to H'3FF0 by a power-on
not initialized by a manual reset or in standby mode. Do not access external memory o
0 until BCR2 register initialization is complete.

Bit:	15	14	13	12	11	10	9
	—	—	A6SZ1	A6SZ0	A5SZ1	A5SZ0	A4SZ
Initial value:	0	0	1	1	1	1	1
R/W:	R	R	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1
	A3SZ1	A3SZ0	A2SZ1	A2SZ0	—	—	—
Initial value:	1	1	1	1	0	0	0
R/W:	R/W	R/W	R/W	R/W	R	R	R

Bits 15, 14, 3, 2, 1, and 0—Reserved: These bits are always read as 0. The write valu
always be 0.

Bits 2n + 1, 2n—Area n (2–6) Bus Size Specification (AnSZ1, AnSZ0): Specify the
physical space area n (n = 2 to 6).

1	Reserved (Setting proh

10.2.3 Wait State Control Register 1 (WCR1)

Wait state control register 1 (WCR1) is a 16-bit readable/writable register that specifies number of idle (wait) state cycles inserted for each area. For some memories, data bus c not be turned off quickly even when the read signal from the external device is turned o can result in conflicts between data buses when consecutive memory accesses are to dif memories or when a write immediately follows a memory read. This LSI automatically number of idle states set in WCR1 in those cases.

WCR1 is initialized to H'3FF3 by a power-on reset. It is not initialized by a manual res standby mode, and retains its contents.

Bit:	15	14	13	12	11	10	9
	WAITSEL	—	A6IW1	A6IW0	A5IW1	A5IW0	A4IW1
Initial value:	0	0	1	1	1	1	1
R/W:	R/W	R	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1
	A3IW1	A3IW0	A2IW1	A2IW0	—	—	A0IW1
Initial value:	1	1	1	1	0	0	1
R/W:	R/W	R/W	R/W	R/W	R	R	R/W

Bits 14, 3, and 2 —Reserved: These bits are always read as 0. The write value should
0.

Bits 2n + 1, 2n—Area n (6–2, 0) Intercycle Idle Specification (AnIW1, AnIW0): S
number of idles inserted between bus cycles when switching between physical space a
0) and another space or between a read access and a write access in the same physical

Bit 2n + 1: AnIW1	Bit 2n: AnIW0	Description
0	0	1 idle cycle inserted
	1	1 idle cycle inserted
1	0	2 idle cycles inserted
	1	3 idle cycles inserted

10.2.4 Wait State Control Register 2 (WCR2)

Wait state control register 2 (WCR2) is a 16-bit readable/writable register that specifie
number of wait state cycles inserted for each area. It also specifies the data access pitc
memory accesses. This allows direct connection of even low-speed memories without
circuit. WCR2 is initialized to H'FFFF by a power-on reset. It is not initialized by a m
or in standby mode.

Bit:	15	14	13	12	11	10	9
	A6 W2	A6 W1	A6 W0	A5 W2	A5 W1	A5 W0	A4 W
Initial value:	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1
	A4 W0	A3 W1	A3 W0	A2 W1	A2 W0	A0 W2	A0 W
Initial value:	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

		1	1	Enabled	2	E
	1	0	2	Enabled	3	E
		1	3	Enabled	4	E
1	0	0	4	Enabled	4	E
		1	6	Enabled	6	E
	1	0	8	Enabled	8	E
		1	10 (Initial value)	Enabled	10	E

Bits 12 to 10—Area 5 Wait Control (A5W2, A5W1, A5W0): Specify the number of inserted in physical space area 5. Also specify the number of states for burst transfer.

			Description			
			First Cycle		Burst Cycle (Excluding First	
Bit 12: A5W2	**Bit 11: A5W1**	**Bit 10: A5W0**	**Inserted Wait States**	**$\overline{\text{WAIT}}$ Pin**	**Number of States Per Data Transfer**	**$\overline{\text{W}}$**
0	0	0	0	Disabled	2	E
		1	1	Enabled	2	E
	1	0	2	Enabled	3	E
		1	3	Enabled	4	E
1	0	0	4	Enabled	4	E
		1	6	Enabled	6	E
	1	0	8	Enabled	8	E
		1	10 (Initial value)	Enabled	10	E

		1	3	Enabled
1	0	0	4	Enabled
		1	6	Enabled
	1	0	8	Enabled
		1	10	Enabled

Bits 6 and 5—Area 3 Wait Control (A3W1, A3W0): Specify the number of wait sta in physical space area 3.

- For Ordinary Memory

		Description	
Bit 6: A3W1	**Bit 5: A3W0**	**Inserted Wait States**	**$\overline{\text{WAIT}}$ Pin**
0	0	0	Ignored
	1	1	Enabled
1	0	2	Enabled
	1	3	Enabled

- For Synchronous DRAM

		Description	
Bit 6: A3W1	**Bit 5: A3W0**	**Synchronous DRAM: CAS Latency**	
0	0	1	
	1	1	
1	0	2	
	1	3	(Initial value)

1	0	2	Enabled	
	1	3	Enabled	(In

- For Synchronous DRAM

		Description	
Bit 4: A2W1	**Bit 3: A2W0**	**Synchronous DRAM: CAS Latency**	
0	0	1	
	1	1	
1	0	2	
	1	3	(Initial value)

Bits 2 to 0—Area 0 Wait Control (A0W2, A0W1, A0W0): Specify the number of wa inserted in physical space area 0. Also specify the burst pitch for burst transfer.

			Description			
			First Cycle		Burst Cycle (Excluding First	
Bit 2: A0W2	**Bit 1: A0W1**	**Bit 0: A0W0**	**Inserted Wait States**	**$\overline{\text{WAIT}}$ Pin**	**Number of States Per Data Transfer**	**W**
0	0	0	0	Ignored	2	En
		1	1	Enabled	2	En
	1	0	2	Enabled	3	En
		1	3	Enabled	4	En
1	0	0	4	Enabled	4	En
		1	6	Enabled	6	En
	1	0	8	Enabled	8	En
		1	10 (Initial value)	Enabled	10	En

modified again. When RFSH and RMODE are written to, write the same values to the
When using synchronous DRAM, do not access areas 2 and 3 until this register is initi

Bit:	15	14	13	12	11	10	9
	TPC1	TPC0	RCD1	RCD0	TRWL1	TRWL0	TRAS
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1
	RASD	AMX3	AMX2	AMX1	AMX0	RFSH	RMOD
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bits 15 and 14—RAS Precharge Time (TPC1, TPC0): When synchronous DRAM
selected as connected memory, they set the minimum number of cycles until output of
bank-active command after precharge. However, the number of cycles input immediat
issue of an all-bank-precharge command (PALL) in the case of an auto-refresh or a pr
command (PRE) in the bank active mode is one fewer than the normal value. TPC1 sh
set to 0 and TPC0 to 1 in the bank active mode.

		Description		
Bit 15: TPC1	**Bit 14: TPC0**	**Normal Operation**	**Immediately after Precharge Command***	**Immediatel Self-Refres**
0	0	1 cycle (Initial value)	0 cycle (Initial value)	2 cycles (I
	1	2 cycles	1 cycle	5 cycles
1	0	3 cycles	2 cycles	8 cycles
	1	4 cycles	3 cycles	11 cycles

Note: * Immediately after all-bank-precharge (PALL) in the case of an auto-refresh or
(PRE) in the bank active mode.

Bits 11 and 10—Write-Precharge Delay (TRWL1, TRWL0): Set the synchronous D write-precharge delay time. This designates the time between the end of a write cycle a bank-active command. This setting is valid only when synchronous DRAM is connecte write cycle, the next bank-active command is not issued for the period TPC + TRWL.

Bit 11: TRWL1	Bit 10: TRWL0	Description	
0	0	1 cycle	(In
	1	2 cycles	
1	0	3 cycles	
	1	Reserved (Setting prohibited)	

Bits 9 and 8—$\overline{\text{CAS}}$-Before-$\overline{\text{RAS}}$ Refresh $\overline{\text{RAS}}$ Assert Time (TRAS1, TRAS0): Whe synchronous DRAM interface is selected, no bank-active command is issued during the TPC + TRAS after an auto-refresh command.

Bit 9: TRAS1	Bit 8: TRAS0	Description	
0	0	2 cycles	(In
	1	3 cycles	
1	0	4 cycles	
	1	5 cycles	

Bit 7—Synchronous DRAM Bank Active (RASD): Specifies whether synchronous D used in bank active mode or auto-precharge mode. Set auto-precharge mode when areas are both designated as synchronous DRAM space.

Bit 7: RASD	Description	
0	Auto-precharge mode	(In
1	Bank active mode	

The bank active mode should not be used unless the bus width for all areas is 32 bits.

		1	0	The row address begins with A11 (The A11 value A1 when the row address is output. 8M × 16-bit × products)*1
0	1	0	0	The row address begins with A9 (The A9 value is when the row address is output. 1M × 16-bit × 4-ba products)
			1	The row address begins with A10 (The A10 value A1 when the row address is output. 2M × 8-bit × 4 products, 2M × 16-bit × 4-bank products)
		1	1	The row address begins with A9 (The A9 value is when the row address is output. 512k × 32-bit × 4- products)*2
0	0	0	0	Begin synchronous DRAM access after setting AM *1**
Except above value				Reserved (Setting prohibited)

Notes: 1. Can only be set when using a 16-bit bus width.
2. Can only be set when using a 32-bit bus width.

Bit 2—Refresh Control (RFSH): The RFSH bit determines whether or not synchron refresh operations are is performed. If the refresh function is not used, the timer for ge periodic refresh requests can also be used as an interval timer.

Bit 2: RFSH	Description
0	No refresh
1	Refresh

0	Auto refresh (RFSH must be 1)	(In
1	Self-refresh (RFSH must be 1)	

Bit 0—Reserved: This bit is always read as 0. The write value should always be 0.

10.2.6 PCMCIA Control Register (PCR)

The PCMCIA control register (PCR) is a 16-bit readable/writable register that specifies assertion and negation timing of the $\overline{OE}$ and $\overline{WE}$ signals for the PCMCIA interface con areas 5 and 6. The $\overline{OE}$ and $\overline{WE}$ signal assertion width is set by the wait control bits in th register.

PCR is initialized to H'0000 by a power-on reset, but is not initialized, and retains its co manual reset and in standby mode.

Bit:	15	14	13	12	11	10	9
	A6W3	A5W3	—	—	A5TED2	A6TED2	A5TEH
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1
	A5TED1	A5TED0	A6TED1	A6TED0	A5TEH1	A5TEH0	A6TEH
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

0	0	0	0	0	Ignored	2
0	0	0	1	1	Enabled	2
0	0	1	0	2	Enabled	3
0	0	1	1	3	Enabled	4
0	1	0	0	4	Enabled	5
0	1	0	1	6	Enabled	7
0	1	1	0	8	Enabled	9
0	1	1	1	10 (Initial value)	Enabled	11
1	0	0	0	12	Enabled	13
1	0	0	1	14	Enabled	15
1	0	1	0	18	Enabled	19
1	0	1	1	22	Enabled	23
1	1	0	0	26	Enabled	27
1	1	0	1	30	Enabled	31
1	1	1	0	34	Enabled	35
1	1	1	1	38	Enabled	39

Bit 14—Area 5 Wait Control (A5W3): Specifies the number of inserted wait states f
combined with bits A5W2–A5W0 in WCR2. Also specifies the number of transfer sta
transfer. Clear this bit to 0 when area 5 is not set to PCMCIA.

The relationship between the set value and the number of waits is the same as for A6W

Bits 13 and 12—Reserved: These bits are always read as 0. The write value should a

		1	3.5-cycle delay
1	0	0	4.5-cycle delay
		1	5.5-cycle delay
	1	0	6.5-cycle delay
		1	7.5-cycle delay

Bits 10, 5, and 4—Area 6 Address $\overline{OE}$/$\overline{WE}$ Assert Delay (A6TED2, A6TED1, A6TE
A6TED bits specify the delay time from address output to $\overline{OE}$/$\overline{WE}$ assertion for the PC
interface connected to area 6.

Bit 10: A6TED2	Bit 5: A6TED1	Bit 4: A6TED0	Description
0	0	0	0.5-cycle delay (In
		1	1.5-cycle delay
	1	0	2.5-cycle delay
		1	3.5-cycle delay
1	0	0	4.5-cycle delay
		1	5.5-cycle delay
	1	0	6.5-cycle delay
		1	7.5-cycle delay

		1	3.5-cycle delay
1	0	0	4.5-cycle delay
		1	5.5-cycle delay
	1	0	6.5-cycle delay
		1	7.5-cycle delay

Bits 8, 1, and 0—Area 6 $\overline{OE}$/$\overline{WE}$ Negate Address Delay (A6TEH2, A6TEH1, A6T
Specify the address hold delay time from $\overline{OE}$/$\overline{WE}$ negation for the PCMCIA interface area 6.

Bit 8: A6TEH2	Bit 1: A6TEH1	Bit 0: A6TEH0	Description
0	0	0	0.5-cycle delay (
		1	1.5-cycle delay
	1	0	2.5-cycle delay
		1	3.5-cycle delay
1	0	0	4.5-cycle delay
		1	5.5-cycle delay
	1	0	6.5-cycle delay
		1	7.5-cycle delay

synchronous DRAM is connected to A2 of the chip and A1 of the synchronous DRAM connected to A3 of the chip, the value actually written to the synchronous DRAM is the shifted two bits right. With a 16-bit bus width, the value written is the X value shifted o right. For example, with a 32-bit bus width, when H'0230 is written to the SDMR regist 2, random data is written to the address H'FFFFD000 (address Y) + H'08C0 (value X), H'FFFFD8C0. As a result, H'0230 is written to the SDMR register. The range for value H'0000 to H'0FFC. When H'0230 is written to the SDMR register of area 3, random dat to the address H'FFFFE000 (address Y) + H'08C0 (value X), or H'FFFFE8C0. As a res is written to the SDMR register. The range for value X is H'0000 to H'0FFC.

Bit:	31	.....	12	11	10	9
	SDMR address			—	—	—
Initial value:	—	.....	—	—	—	—
R/W:	—	.....	—	W*	W*	W

Bit:	7	6	5	4	3	2	1
	—	—	—	—	—	—	—
Initial value:	—	—	—	—	—	—	—
R/W:	W	W	W	W	W	W	—

Note: * Depending on the type of synchronous DRAM.

B'10100101 and the lower byte as the write data. For details, see section 10.2.
on Accessing Refresh Control Related Registers.

Bit:	15	14	13	12	11	10	9
	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R

Bit:	7	6	5	4	3	2	1
	CMF	CMIE	CKS2	CKS1	CKS0	OVF	OVI
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bits 15 to 8—Reserved: These bits are always read as 0. The write value should alwa

Bit 7—Compare Match Flag (CMF): Indicates that the values of RTCNT and RTCO

Bit 7: CMF	Description
0	The values of RTCNT and RTCOR do not match (
	Clearing condition: When a refresh is performed after 0 has bee CMF and RFSH = 1 and RMODE = 0 (to perform a CBR refresh
1	The values of RTCNT and RTCOR match
	Setting condition: RTCNT = RTCOR*

Note: * Contents do not change when 1 is written to CMF.

Bit 6—Compare Match Interrupt Enable (CMIE): Enables or disables an interrupt caused when CMF in RTCSR is set to 1. Do not set this bit to 1 when using auto-refres

Bit 6: CMIE	Description
0	Interrupt request by CMF is disabled (
1	Interrupt request by CMF is enabled

	1	0	CKIO/16
		1	CKIO/64
1	0	0	CKIO/256
		1	CKIO/1024
	1	0	CKIO/2048
		1	CKIO/4096

Bit 2—Refresh Count Overflow Flag (OVF): Indicates when the number of refresh r
indicated in the refresh count register (RFCR) exceeds the limit set in the LMTS bit in

Bit 2: OVF	Description
0	RFCR has not exceeded the count limit value set in LMTS (In Clearing condition: When 0 is written to OVF
1	RFCR has exceeded the count limit value set in LMTS Setting condition: When the RFCR value has exceeded the count set in LMTS*

Note: * Contents do not change when 1 is written to OVF.

Bit 1—Refresh Count Overflow Interrupt Enable (OVIE): Selects whether to suppr
generation of interrupt requests by the OVF bit in RTCSR when OVF is set to 1.

Bit 1: OVIE	Description
0	Interrupt request by OVF is disabled (In
1	Interrupt request by OVF is enabled

10.2.9 Refresh Timer Counter (RTCNT)

RTCNT is a 16-bit register containing a readable/writable 8-bit counter that counts up
clock. The clock select bits (CKS2–CKS0) in RTCSR select the input clock. When RT
matches RTCOR, the CMF bit in RTCSR is set and RTCNT is cleared. RTCNT is init
H'00 by a power-on reset, but continues incrementing after a manual reset. It is not ini
standby mode, but holds its contents.

Note: The method of writing to RTCNT differs from that for general registers to ens
RTCNT is not rewritten incorrectly. Use a word transfer instruction to set the
as B'10100101 and the lower byte as the write data. For details, see section 10
Cautions on Accessing Refresh Control Related Registers.

Bit:	15	14	13	12	11	10	9
Initial value:	0	0	0	0	0	0	0
R/W:	—	—	—	—	—	—	—

Bit:	7	6	5	4	3	2	1
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

bits CKS2 to CKS0 in RTCSR.

Note: The method of writing to RTCOR differs from that for general registers to ensu RTCOR is not rewritten incorrectly. Use a word transfer instruction to set the u as B'10100101 and the upper byte as the write data. For details, see section 10.2 Cautions on Accessing Refresh Control Related Registers.

Bit:	15	14	13	12	11	10	9
Initial value:	0	0	0	0	0	0	0
R/W:	—	—	—	—	—	—	—

Bit:	7	6	5	4	3	2	1
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

10.2.11 Refresh Count Register (RFCR)

The refresh count register (RFCR) counts the number of refreshing. When RFCR excee count limit value set in the LMTS bit in RTCSR, the OVF bit in RTCSR is set and RFC cleared. RFCR is a 10-bit readable/writable counter. RFCR is initialized to H'0000 by a reset. RFCR continues incrementing in a manual reset. It is not initialized by in standby holds its contents.

Note: The method of writing to RFCR differs from that for general registers to ensure is not rewritten incorrectly. Use a word transfer instruction to set the six bits sta the MSB in the upper byte as B'101001, and the remaining bits as the write data details, see section 10.2.12, Cautions on Accessing Refresh Control Related Re

R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

10.2.12 Cautions on Accessing Refresh Control Related Registers

RFCR, RTCSR, RTCNT, and RTCOR require that a specific code be appended to the
is written to prevent data from being mistakenly overwritten by program overruns or
operations (figure 10.5). Perform reads and writes using the following methods:

1. When writing to RFCR, RTCSR, RTCNT, and RTCOR, use only word transfer ins
   Byte transfer instructions cannot be used.

   When writing to RTCNT, RTCSR, or RTCOR, place B'10100101 in the upper byt
   write data in the lower byte. When writing to RFCR, place B'101001 in the upper
   the write data in the remaining bits, as shown in figure 10.5.
2. When reading from RFCR, RTCSR, RTCNT, and RTCOR, carry out reads with a
   width. 0 is read from undefined bits.

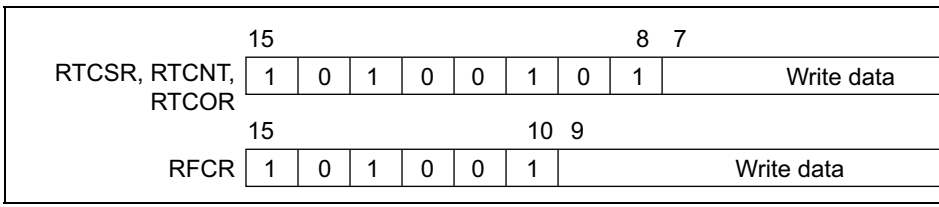


Figure 10.5 Writing to RFCR, RTCSR, RTCNT, and RTCOR

PC0MD[1:0] in the PCCR register should be set to 00 (other function).

Bit:	15	14	13	12	11	10	9
	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R

Bit:	7	6	5	4	3	2	1
	—	CS2/0	CAP1	CAP0	A25	A24	A23
Initial value:	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W

Bits 15 to 7—Reserved: These bits are always read as 0. The write value should alway

Bit 6—CS2/CS0 Select (CS2/0): Selects whether an area 2 or area 0 address is to be d

Bit 6: CS2/0	Description
0	Area 0 is selected
1	Area 2 is selected

Only 0 should be used for the CS2/0 bit in MCSCR0. Either 0 or 1 may be used for MC MCSCR7.

Bits 5 and 4—Connected Memory Size Specification (CAP1, CAP0)

Bit 5: CAP1	Bit 4: CAP0	Description
0	0	32-Mbit memory is connected
0	1	64-Mbit memory is connected
1	0	128-Mbit memory is connected
1	1	256-Mbit memory is connected

Bits 3 to 0—Start Address Specification (A25, A24, A23, A22): These bits specify th address of the memory area for which $\overline{\text{MCS}[0]}$ is asserted.

The bit configuration and functions are the same as those of MCSCR0.

10.2.16 MCS3 Control Register (MCSCR3)

The MCS3 control register (MCSCR3) specifies the $\overline{\text{MCS}}[3]$ pin output conditions.

The bit configuration and functions are the same as those of MCSCR0.

10.2.17 MCS4 Control Register (MCSCR4)

The MCS4 control register (MCSCR4) specifies the $\overline{\text{MCS}}[4]$ pin output conditions.

The bit configuration and functions are the same as those of MCSCR0.

10.2.18 MCS5 Control Register (MCSCR5)

The MCS5 control register (MCSCR5) specifies the $\overline{\text{MCS}}[5]$ pin output conditions.

The bit configuration and functions are the same as those of MCSCR0.

10.2.19 MCS6 Control Register (MCSCR6)

The MCS6 control register (MCSCR6) specifies the $\overline{\text{MCS}}[6]$ pin output conditions.

The bit configuration and functions are the same as those of MCSCR0.

10.2.20 MCS7 Control Register (MCSCR7)

The MCS7 control register (MCSCR7) specifies the $\overline{\text{MCS}}[7]$ pin output conditions.

The bit configuration and functions are the same as those of MCSCR0.

Three data bus widths are available for ordinary memory (byte, word, longword) and tw
widths (word and longword) for synchronous DRAM. For the PCMCIA interface, choo
byte and word. This means data alignment is done by matching the device's data width
endian. The access unit must also be matched to the device's bus width. This also mean
longword data is read from a byte-width device, four read operations must be performe
SH7709S, data alignment and conversion of data length is performed automatically bet
respective interfaces.

Tables 10.7 to 10.12 show the relationship between endian, device data width, and acce

Table 10.7 32-Bit External Device/Big-Endian Access and Data Alignment

	Data Bus				Strobe Signals		
Operation	**D31–D24**	**D23–D16**	**D15–D8**	**D7–D0**	**$\overline{WE3}$, DQMUU**	**$\overline{WE2}$, DQMUL**	**$\overline{WE1}$, DQMLU**
Byte access at 0	Data 7–0	—	—	—	Asserted		
Byte access at 1	—	Data 7–0	—	—		Asserted	
Byte access at 2	—	—	Data 7–0	—			Asserted
Byte access at 3	—	—	—	Data 7–0			
Word access at 0	Data 15–8	Data 7–0	—	—	Asserted	Asserted	
Word access at 2	—	—	Data 15–8	Data 7–0			Asserted
Longword access at 0	Data 31–24	Data 23–16	Data 15–8	Data 7–0	Asserted	Asserted	Asserted

Byte access at 2		—	—	Data 7–0	—	Asserted
Byte access at 3		—	—	—	Data 7–0	
Word access at 0		—	—	Data 15–8	Data 7–0	Asserted
Word access at 2		—	—	Data 15–8	Data 7–0	Asserted
Longword access at 0	1st time at 0	—	—	Data 31–24	Data 23–16	Asserted
	2nd time at 2	—	—	Data 15–8	Data 7–0	Asserted

Byte access at 3					Data 7–0
Word access at 0	1st time at 0	—	—	—	Data 15–8
	2nd time at 1	—	—	—	Data 7–0
Word access at 2	1st time at 2	—	—	—	Data 15–8
	2nd time at 3	—	—	—	Data 7–0
Longword access at 0	1st time at 0	—	—	—	Data 31–24
	2nd time at 1	—	—	—	Data 23–16
	3rd time at 2	—	—	—	Data 15–8
	4th time at 3	—	—	—	Data 7–0

Byte access at 2	—	Data 7–0	—	—		Asserted	
Byte access at 3	Data 7–0	—	—	—	Asserted		
Word access at 0	—	—	Data 15–8	Data 7–0			Asserted
Word access at 2	Data 15–8	Data 7–0	—	—	Asserted	Asserted	
Longword access at 0	Data 31–24	Data 23–16	Data 15–8	Data 7–0	Asserted	Asserted	Asserted

Table 10.11 16-Bit External Device/Little-Endian Access and Data Alignment

		Data Bus				Strobe Signals		
Operation		**D31–D24**	**D23–D16**	**D15–D8**	**D7–D0**	**$\overline{WE3}$, DQMUU**	**$\overline{WE2}$, DQMUL**	**$\overline{WE1}$, DQMLU**
Byte access at 0		—	—	—	Data 7–0			
Byte access at 1		—	—	Data 7–0	—			Asserted
Byte access at 2		—	—	—	Data 7–0			
Byte access at 3		—	—	Data 7–0	—			Asserted
Word access at 0		—	—	Data 15–8	Data 7–0			Asserted
Word access at 2		—	—	Data 15–8	Data 7–0			Asserted
Longword access at 0	1st time at 0	—	—	Data 15–8	Data 7–0			Asserted
	2nd time at 2	—	—	Data 31–24	Data 23–16			Asserted

Byte access at 2		—	—	—	Data 7–0
Byte access at 3		—	—	—	Data 7–0
Word access at 0	1st time at 0	—	—	—	Data 7–0
	2nd time at 1	—	—	—	Data 15–8
Word access at 2	1st time at 2	—	—	—	Data 7–0
	2nd time at 3	—	—	—	Data 15–8
Longword access at 0	1st time at 0	—	—	—	Data 7–0
	2nd time at 1	—	—	—	Data 15–8
	3rd time at 2	—	—	—	Data 23–16
	4th time at 3	—	—	—	Data 31–24

WE0–WE3 signals for write control are also asserted. The number of bus cycles is sel
between 0 and 10 wait cycles using the A0W2–A0W0 bits in WCR2. When the burst
used, the bus cycle pitch of the burst cycle is determined within a range of 2–10 accor
number of waits.

Area 1: Area 1 physical address bits A28–A26 are 001. Address bits A31–A29 are ig
the address range is H'04000000 + H'20000000 × n – H'07FFFFFF + H'20000000 × n
and n = 1–6 are the shadow spaces).

Area 1 is the area specifically for internal peripheral modules. External memories can
connected.

Control registers of the peripheral modules shown below are mapped to this area 1. Th
addresses are physical addresses, to which logical addresses can be mapped when the
enabled:

DMAC, PORT, IrDA, SCIF, ADC, DAC, INTC (except INTEVT, IPRA, IPRB)

These registers must be set not to be cached by using software.

Area 2: Area 2 physical address bits A28–A26 are 010. Address bits A31–A29 are ig
the address range is H'08000000 + H'20000000 × n – H'0BFFFFFF + H'20000000 × n
and n = 1–6 are the shadow spaces).

Ordinary memories such as SRAM and ROM, as well as synchronous DRAM, can be
to this space. Byte, word, or longword can be selected as the bus width using bits A2S
A2SZ0 in BCR2 for ordinary memory.

When the area 2 space is accessed, the $\overline{\text{CS2}}$ signal is asserted. When ordinary memori
connected, the $\overline{\text{RD}}$ signal that can be used as $\overline{\text{OE}}$ and the $\overline{\text{WE0}}$–$\overline{\text{WE3}}$ signals for write
also asserted and the number of bus cycles is selected between 0 and 3 wait cycles usi
A2W1 and A2W0 bits in WCR2. Only when ordinary memories are connected, any w
inserted in each bus cycle by means of the external wait pin ($\overline{\text{WAIT}}$).

When synchronous DRAM is connected, the $\overline{\text{RAS3U}}$ and $\overline{\text{RAS3L}}$ signals, $\overline{\text{CASU}}$ and
signals, RD/$\overline{\text{WR}}$ signal, and byte control signals DQMHH, DQMHL, DQMLH, and D
all asserted and addresses multiplexed. Control of $\overline{\text{RAS3U}}$, $\overline{\text{RAS3L}}$, $\overline{\text{CASU}}$, $\overline{\text{CASL}}$, da
and address multiplexing is set with MCR.

When ordinary memories are connected, the $\overline{RD}$ signal that can be used as $\overline{OE}$ and the $\overline{W}$ signals for write control are asserted and the number of bus cycles is selected between 0 cycles using the A3W1 and A3W0 bits in WCR2.

When synchronous DRAM is connected, the $\overline{RAS3U}$ and $\overline{RAS3L}$ signals, $\overline{CASU}$ and $\overline{C}$ signals, RD/$\overline{WR}$ signal, and byte control signals DQMHH, DQMHL, DQMLH, and DQ all asserted and addresses multiplexed.

Area 4: Area 4 physical address bits A28–A26 are 100. Address bits A31–A29 are ignc the address range is H'10000000 + H'20000000 × n – H'13FFFFFF + H'20000000 × n (and n = 1–6 are the shadow spaces).

Only ordinary memories such as SRAM and ROM can be connected to this space. Byte longword can be selected as the bus width using bits A4SZ1 and A4SZ0 in BCR2. Whe 4 space is accessed, the $\overline{CS4}$ signal is asserted. The $\overline{RD}$ signal that can be used as $\overline{OE}$ a $\overline{WE0}$–$\overline{WE3}$ signals for write control are also asserted. The number of bus cycles is selec between 0 and 10 wait cycles using the A4W2–A4W0 bits in WCR2. Any wait can be i each bus cycle by means of the external wait pin ($\overline{WAIT}$).

Area 5: Area 5 physical address bits A28–A26 are 101. Address bits A31–A29 are ignc the address range is the 64 Mbytes at H'14000000 + H'20000000 × n – H'17FFFFFF + H'20000000 × n (n = 0–6 and n = 1–6 are the shadow spaces).

Ordinary memories such as SRAM and ROM as well as burst ROM and PCMCIA inter be connected to this space. When the PCMCIA interface is used, the IC memory card i address range comprises the 32 Mbytes at H'14000000 + H'20000000 × n to H'15FFFF H'20000000 × n (n = 0–6 and n = 1–6 are the shadow spaces), and the I/O card interfac range comprises the 32 Mbytes at H'16000000 + H'20000000 × n to H'17FFFFFF + H'2 n (n = 0–6 and n = 1–6 are the shadow spaces).

For ordinary memory and burst ROM, byte, word, or longword can be selected as the b using bits A5SZ1 and A5SZ0 in BCR2. For the PCMCIA interface, byte or word can b as the bus width using bits A5SZ1 and A5SZ0 bits in BCR2.

bus cycle pitch of the burst cycle is determined within a range of 2–11 (2–39 for the P
interface) according to the number of waits. The setup and hold times of address/$\overline{CS5}$
read/write strobe signals can be set in the range 0.5–7.5 using bits A5TED2–A5TED0
A5TEH2–A5TEH0 in the PCR register.

Area 6: Area 6 physical address bits A28–A26 are 110. Address bits A31–A29 are ig
the address range is the 64 Mbytes at H'18000000 + H'20000000 × n – H'1BFFFFFF +
H'20000000 × n (n = 0–6 and n = 1–6 are the shadow spaces).

Ordinary memories such as SRAM and ROM as well as burst ROM and PCMCIA int
be connected to this space. When the PCMCIA interface is used, the IC memory card
address range is 32 Mbytes at H'18000000 + H'20000000 × n – H'19FFFFFF + H'200
and the I/O card interface address range is 32 Mbytes at H'1A000000 + H'20000000 ×
H'1BFFFFFF + H'20000000 × n (n = 0–6 and n = 1–6 are the shadow spaces).

For ordinary memory and burst ROM, byte, word, or longword can be selected as the
using bits A6SZ1 and A6SZ0 in BCR2. For the PCMCIA interface, byte or word can
as the bus width using bits A6SZ1 and A6SZ0 in BCR2.

When the area 6 space is accessed and ordinary memory is connected, the $\overline{CS6}$ signal
The $\overline{RD}$ signal that can be used as $\overline{OE}$ and the $\overline{WE0}$–$\overline{WE3}$ signals for write control are
asserted. When the PCMCIA interface is used, the $\overline{CE1B}$ signal, $\overline{CE2B}$ signal, $\overline{RD}$ sig
signal, and $\overline{WE}$, $\overline{ICIORD}$, and $\overline{ICIOWR}$ signals are asserted.

The number of bus cycles is selected between 0 and 10 wait cycles using the A6W2–A
WCR2. With the PCMCIA interface, from 0 to 38 wait cycles can be selected using th
A6W0 bits in WCR2 and the A6W3 bit in PCR. In addition, any number of waits can
in each bus cycle by means of the external wait pin ($\overline{WAIT}$). The bus cycle pitch of th
is determined within a range of 2–11 (2–39 for the PCMCIA interface) according to th
waits. The address/$\overline{CS6}$ setup and hold times for the read/write strobe signals can be s
range 0.5–7.5 using bits A6TED2–A6TED0 and A6TEH2–A6TEH0 in the PCR regis

There is no access size specification when reading. The correct access start address is o
least significant bit of the address, but since there is no access size specification, 32 bits
read in case of a 32-bit device, and 16 bits in case of a 16-bit device. When writing, onl
signal for the byte to be written is asserted. For details, see section 10.3.1, Endian/Acce
Data Alignment.

Read/write for cache fill or write-back follows the set bus width and transfers a total of
continuously. The bus is not released during this transfer. For cache misses that occur d
or word operand accesses or branching to odd word boundaries, the fill is always perfo
longword accesses on the chip-external interface. Write-through-area write access and
cacheable read/write access are based on the actual address size.

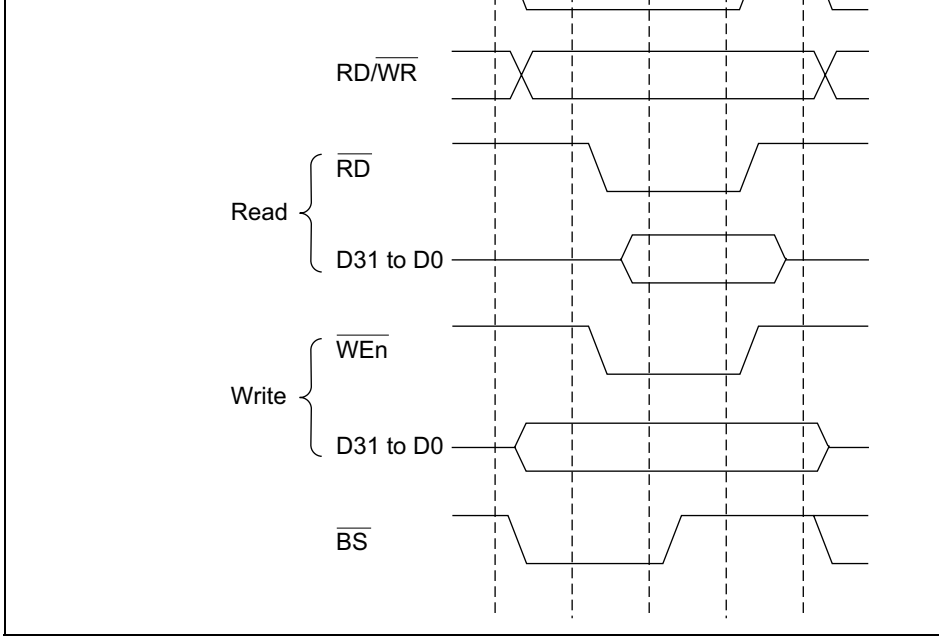


Figure 10.6 Basic Timing of Basic Interface

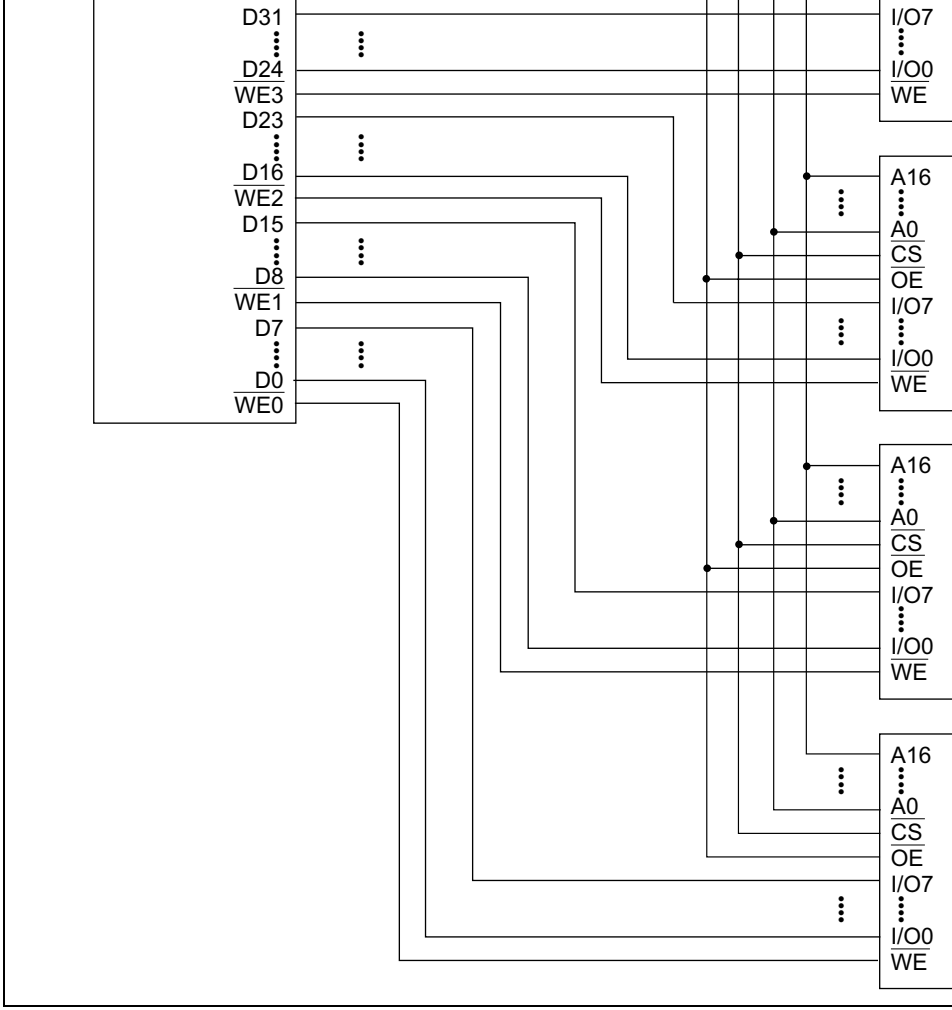


Figure 10.7 Example of 32-Bit Data-Width Static RAM Connection

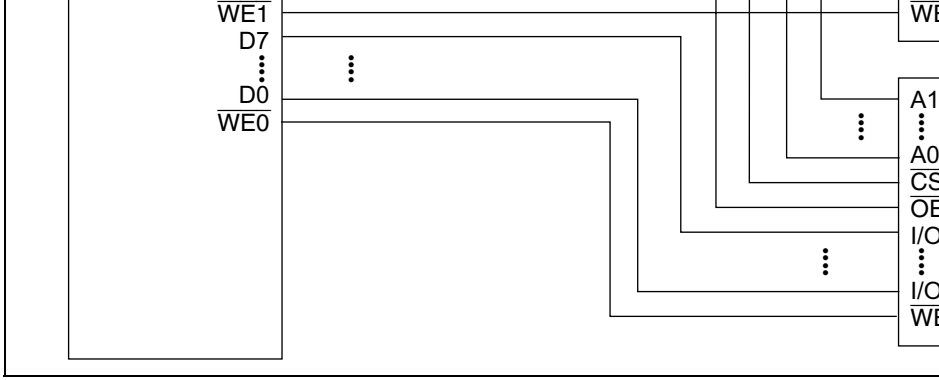


Figure 10.8 Example of 16-Bit Data-Width Static RAM Connection

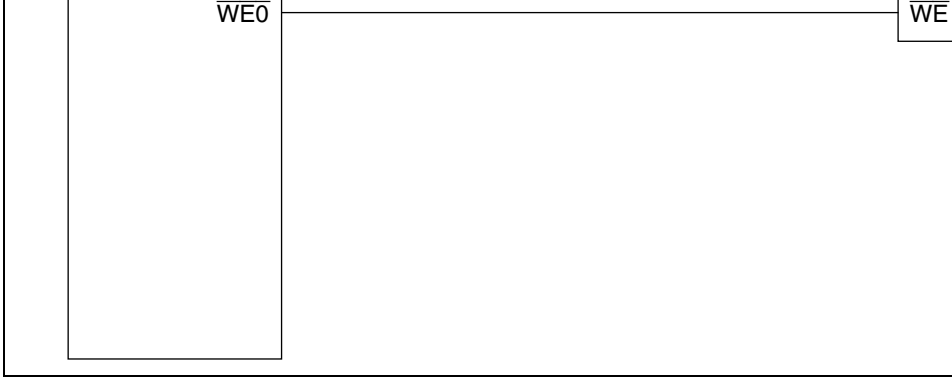


Figure 10.9 Example of 8-Bit Data-Width Static RAM Connection

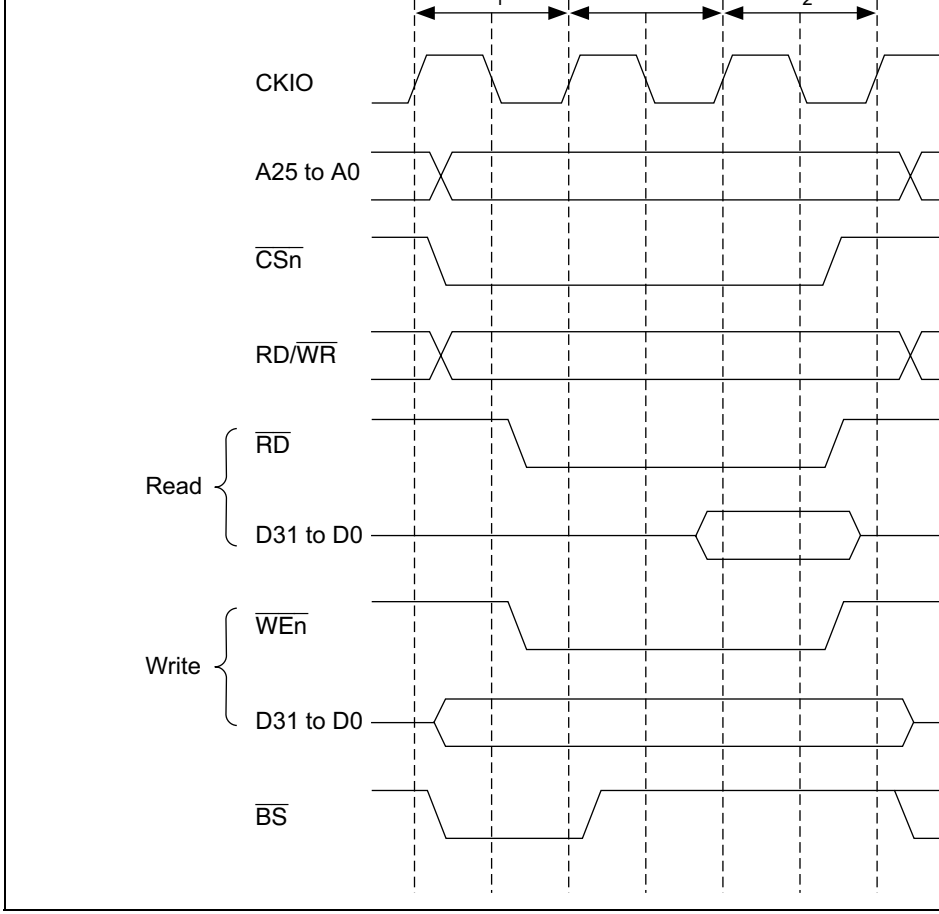


Figure 10.10 Basic Interface Wait Timing (Software Wait Only)

However, the WAIT signal is ignored in the following three cases:

- A write to external address space in dual address mode with 16-byte DMA transfer
- Transfer from an external device with DACK to external address space in single ad with 16-byte DMA transfer
- Cache write-back access

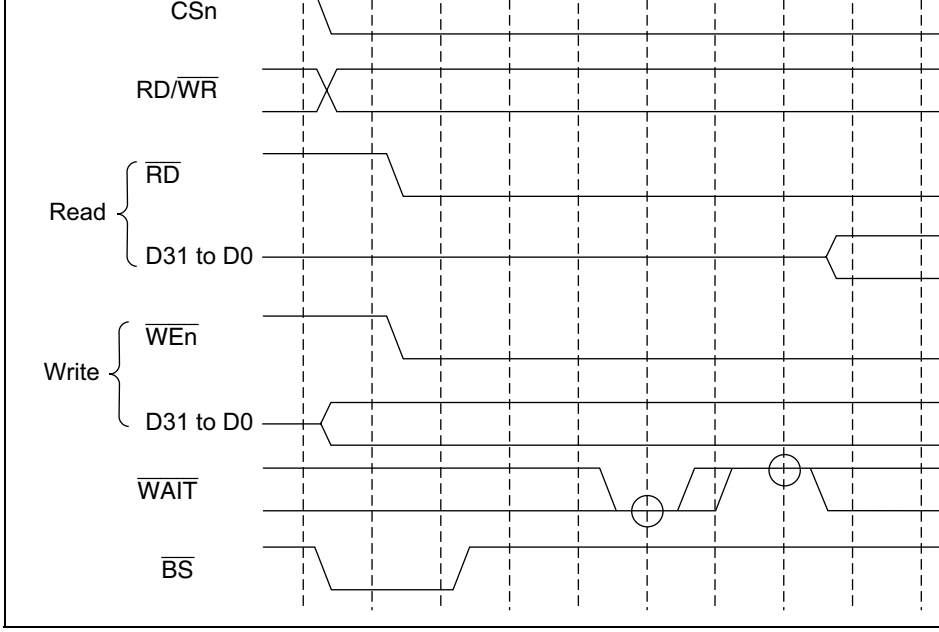


Figure 10.11 Basic Interface Wait State Timing (Wait State Insertion by WA WAITSEL = 1)

With the SH7709S, burst length 1 burst read/single write mode is supported as the synch
DRAM operating mode. A data bus width of 16 or 32 bits can be selected. A 16-bit bur
is performed in a cache fill/write-back cycle, and only one access is performed in a writ
area write or a non-cacheable area read/write.

The control signals for direct connection of synchronous DRAM are $\overline{\text{RAS3L}}$, $\overline{\text{RAS3U}}$,
$\overline{\text{CASU}}$, RD/$\overline{\text{WR}}$, $\overline{\text{CS2}}$ or $\overline{\text{CS3}}$, DQMUU, DQMUL, DQMLU, DQMLL, and CKE. All t
other than $\overline{\text{CS2}}$ and $\overline{\text{CS3}}$ are common to all areas, and signals other than CKE are valid
to the synchronous DRAM only when $\overline{\text{CS2}}$ or $\overline{\text{CS3}}$ is asserted. Synchronous DRAM can
be connected in parallel to a number of areas. CKE is negated (low) only when self-refr
performed, and is always asserted (high) at other times.

In the refresh cycle and mode-register write cycle, $\overline{\text{RAS3U}}$ and $\overline{\text{RAS3L}}$ or $\overline{\text{CASU}}$ and $\overline{\text{C}}$
output.

Commands for synchronous DRAM are specified by $\overline{\text{RAS3L}}$, $\overline{\text{RAS3U}}$, $\overline{\text{CASL}}$, $\overline{\text{CASU}}$,
and special address signals. The commands are NOP, auto-refresh (REF), self-refresh (
precharge all banks (PALL), row address strobe bank active (ACTV), read (READ), rea
precharge (READA), write (WRIT), write with precharge (WRITA), and mode register
(MRS).

Byte specification is performed by DQMUU, DQMUL, DQMLU, and DQMLL. A read
performed for the byte for which the corresponding DQM is low. In big-endian mode, D
specifies an access to address 4n, and DQMLL specifies an access to address 4n + 3. In
endian mode, DQMUU specifies an access to address 4n + 3, and DQMLL specifies an
address 4n.

Figures 10.12 and 10.13 show examples of the connection of two 1M × 16-bit × 4-bank
synchronous DRAMs and one 1M × 16-bit × 4-bank synchronous DRAM, respectively

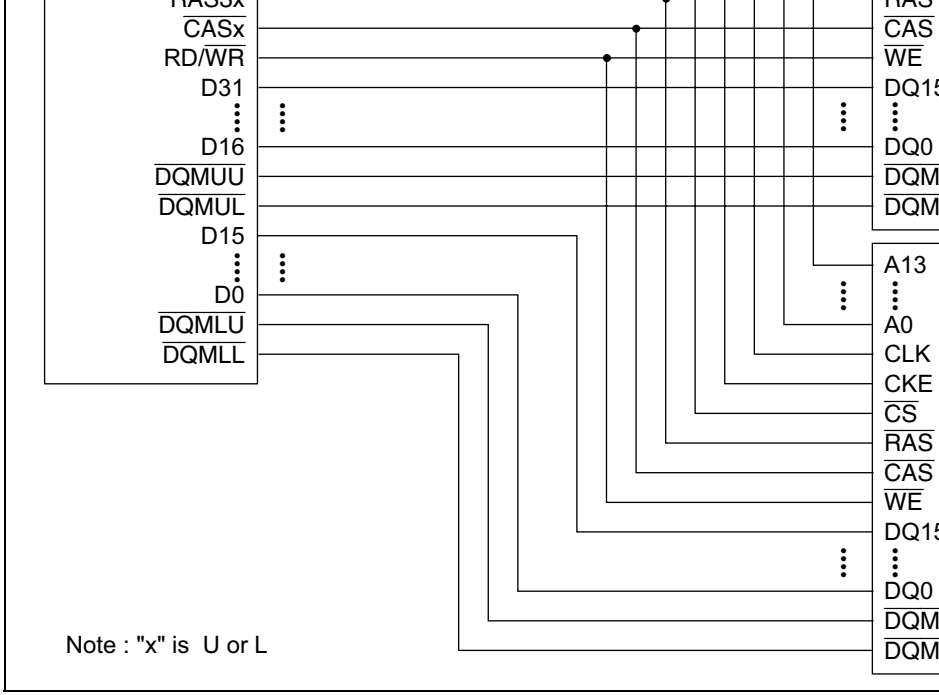


Figure 10.12 Example of 64-Mbit Synchronous DRAM Connection (32-Bit Bu

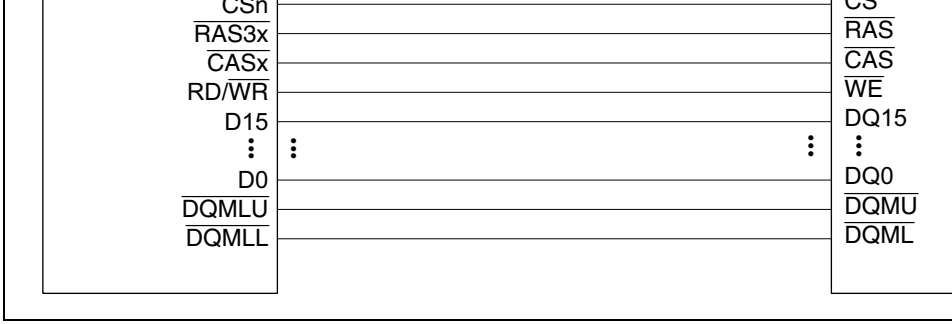


Figure 10.13 Example of 64-Mbit Synchronous DRAM Connection (16-Bit Bus

Address Multiplexing: Synchronous DRAM can be connected without external multip circuitry in accordance with the address multiplex specification bits AMX2-AMX0 in M 10.13 shows the relationship between the address multiplex specification bits and the bi the address pins.

A25–A17 and A0 are not multiplexed; the original values are always output at these pin

When A0, the LSB of the synchronous DRAM address, is connected to the SH7709S, i longword address specification. Connection should therefore be made in the following a 32-bit bus width, connect pin A0 of the synchronous DRAM to pin A2 of the SH7709 connect pin A1 to pin A3; with a 16-bit bus width, connect pin A0 of the synchronous I pin A1 of the SH7709S, then connect pin A1 to pin A2.

						address	A17						
	2M × 16bits × 4banks*2	0	1	0	1	Column address	A1 to A8	A9	A10	A11	L/H*3	A13	A23
						Row address	A10 to A17	A18	A19	A20	A21	A22	A23
	1M × 16bits × 4banks*2	0	1	0	0	Column address	A1 to A8	A9	A10	A11	L/H*3	A13	A22
						Row address	A9 to A16	A17	A18	A19	A20	A21	A22
	2M × 8bits × 4banks*2	0	1	0	1	Column address	A1 to A8	A9	A10	A11	L/H*3	A13	A23
						Row address	A10 to A17	A18	A19	A20	A21	A22	A23
	512k × 32bits × 4banks*2	0	1	1	1	Column address	A1 to A8	A9	A10	A11	L/H*3	A21*4	A22
						Row address	A9 to A16	A17	A18	A19	A20	A21*4	A22
16 bits	8M × 16bits × 4banks*1	1	1	1	0	Column address	A1 to A8	A9	A10	L/H*3	A12	A23	A24
						Row address	A11 to A18	A19	A20	A21	A22	A23	A24
	4M × 16bits × 4banks*2	1	1	0	1	Column address	A1 to A8	A9	A10	L/H*3	A12	A22	A23
						Row address	A10 to A17	A18	A19	A20	A21	A22	A23

	16bits × 4banks[*2]					address	A8						
						Row address	A 9 to A16	A17	A18	A19	A20	A21[*4]	A22[*
	2M × 8bits × 4banks[*2]	0	1	0	1	Column address	A1 to A8	A9	A10	L/H[*3]	A12	A22[*4]	A23[*
						Row address	A10 to A17	A18	A19	A20	A21	A22[*4]	A23[*

Notes: 1. Only RAL3L or CASL is output.

2. When addresses are upper 32 Mbytes, $\overline{\text{RAS3U}}$ or $\overline{\text{CASU}}$ is output.
   When addresses are lower 32 Mbytes, $\overline{\text{RAS3L}}$ or $\overline{\text{CASL}}$ is output.
3. L/H is a bit used in the command specification; it is fixed at L or H according access mode.
4. Bank address specification

A12	A20	L/H	A10	Address precharg
A11	A19	A11	A9	Address
A10	A18	A10	A8	
A9	A17	A9	A7	
A8	A16	A8	A6	
A7	A15	A7	A5	
A6	A14	A6	A4	
A5	A13	A5	A3	
A4	A12	A4	A2	
A3	A11	A3	A1	
A2	A10	A2	A0	
A1	A9	A1	Not used	
A0	A0	A0	Not used	

Burst Read: In the example in figure 10.15 it is assumed that four 2M × 8-bit synchro DRAMs are connected and a 32-bit data width is used, and the burst length is 1. Follo cycle in which ACTV command output is performed, a READ command is issued in t and Tc3 cycles, and a READA command in the Tc4 cycle, and the read data is accepte rising edge of the external command clock (CKIO) from cycle Td1 to cycle Td4. The used to wait for completion of auto-precharge based on the READA command inside synchronous DRAM; no new access command can be issued to the same bank during but access to synchronous DRAM for another area is possible. In the SH7709S, the nu cycles is determined by the TPC bit specification in MCR, and commands cannot be is same synchronous DRAM during this interval.

The example in figure 10.14 shows the basic cycle. To connect low-speed synchronou the cycle can be extended by setting WCR2 and MCR bits. The number of cycles from command output cycle, Tr, to the READ command output cycle, Tc1, can be specified RCD bits in MCR, with values of 0 to 3 specifying 1 to 4 cycles, respectively. In case cycles, a Trw cycle, in which an NOP command is issued for the synchronous DRAM between the Tr cycle and the Tc cycle. The number of cycles from READ and READA output cycles Tc1-Tc4 to the first read data latch cycle, Td1, can be specified as 1 to 3

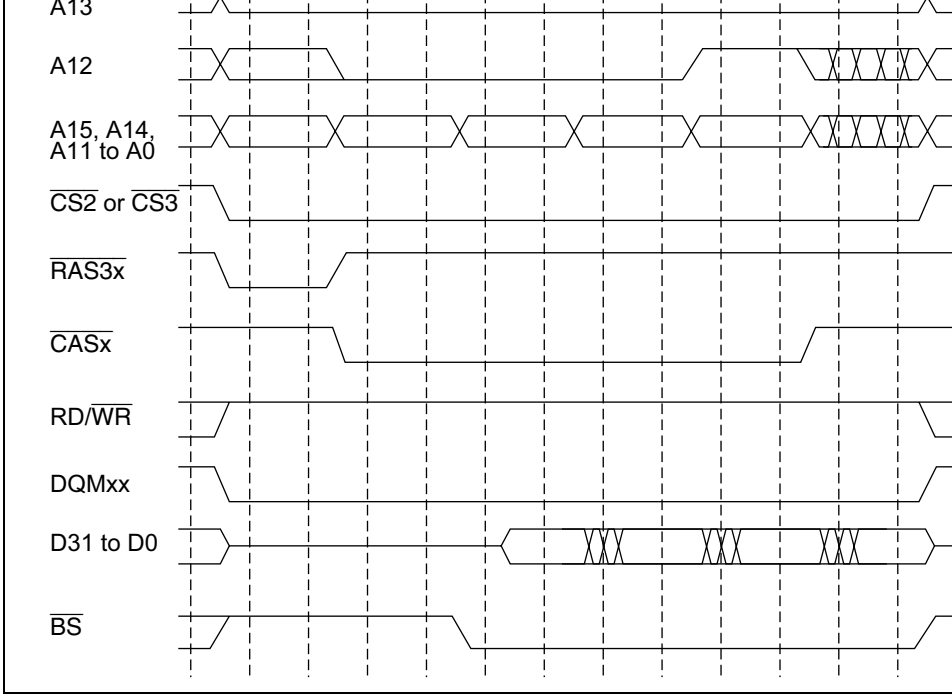


Figure 10.14 Basic Timing for Synchronous DRAM Burst Read

missed data is read in wraparound mode.

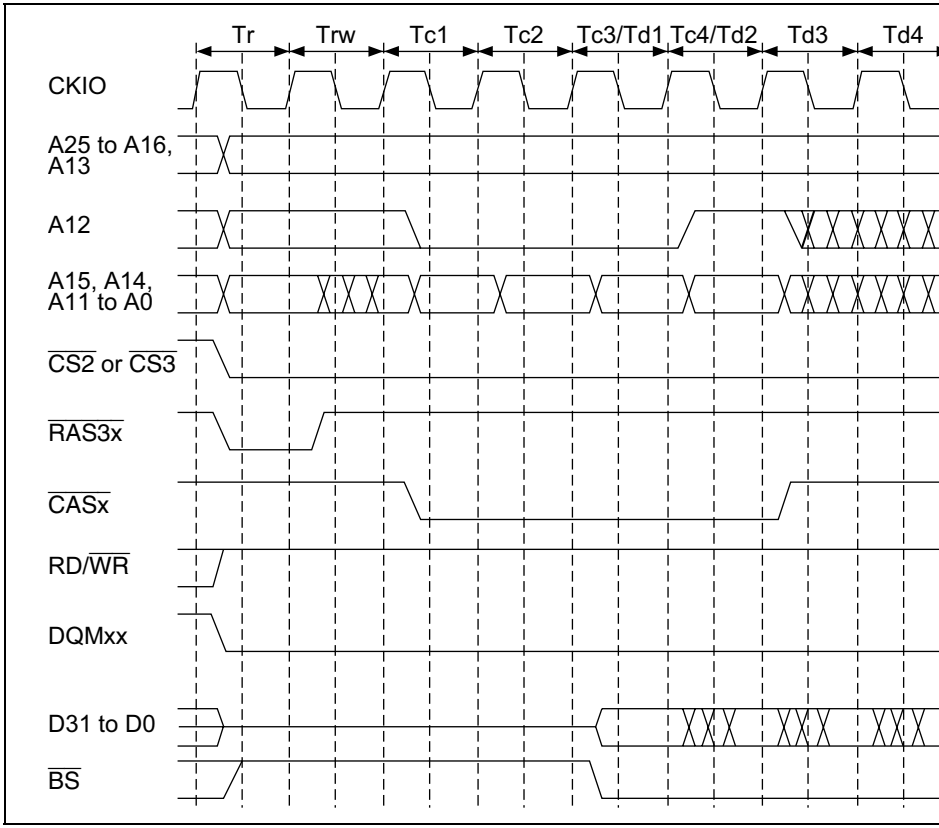


Figure 10.15 Synchronous DRAM Burst Read Wait Specification Tim

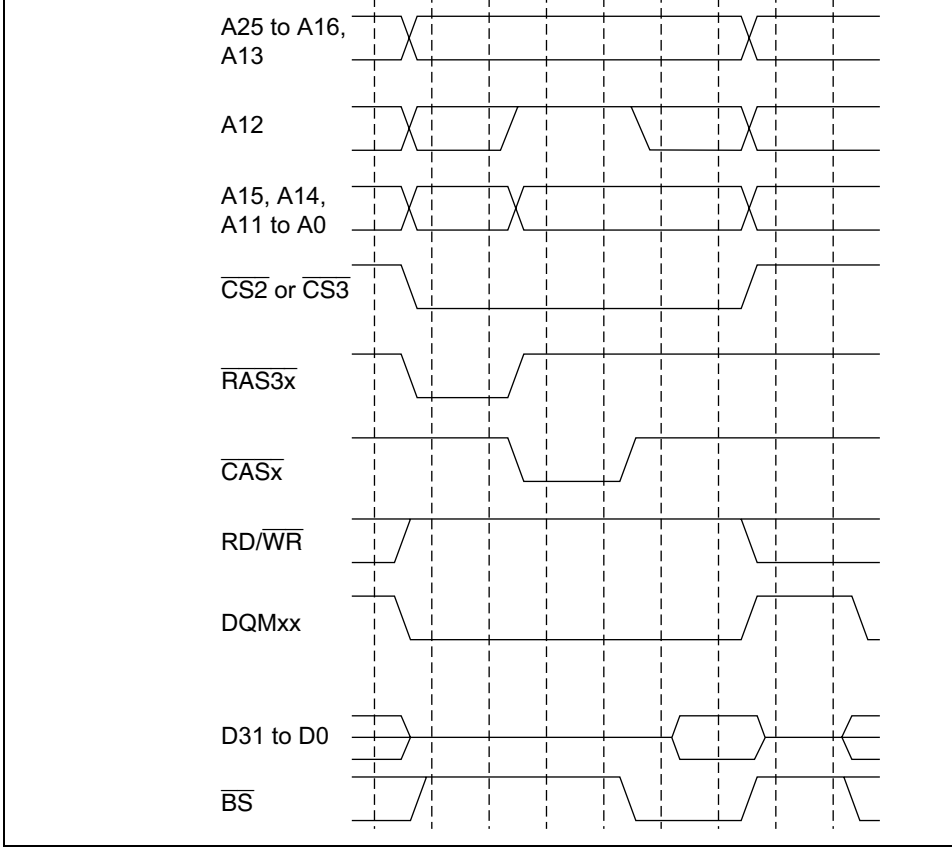


Figure 10.16 Basic Timing for Synchronous DRAM Single Read

also added as a wait interval until precharging is started following the write command
a new command for the same bank is deferred during this interval. The number of Trw
be specified by the TRWL bits in MCR.

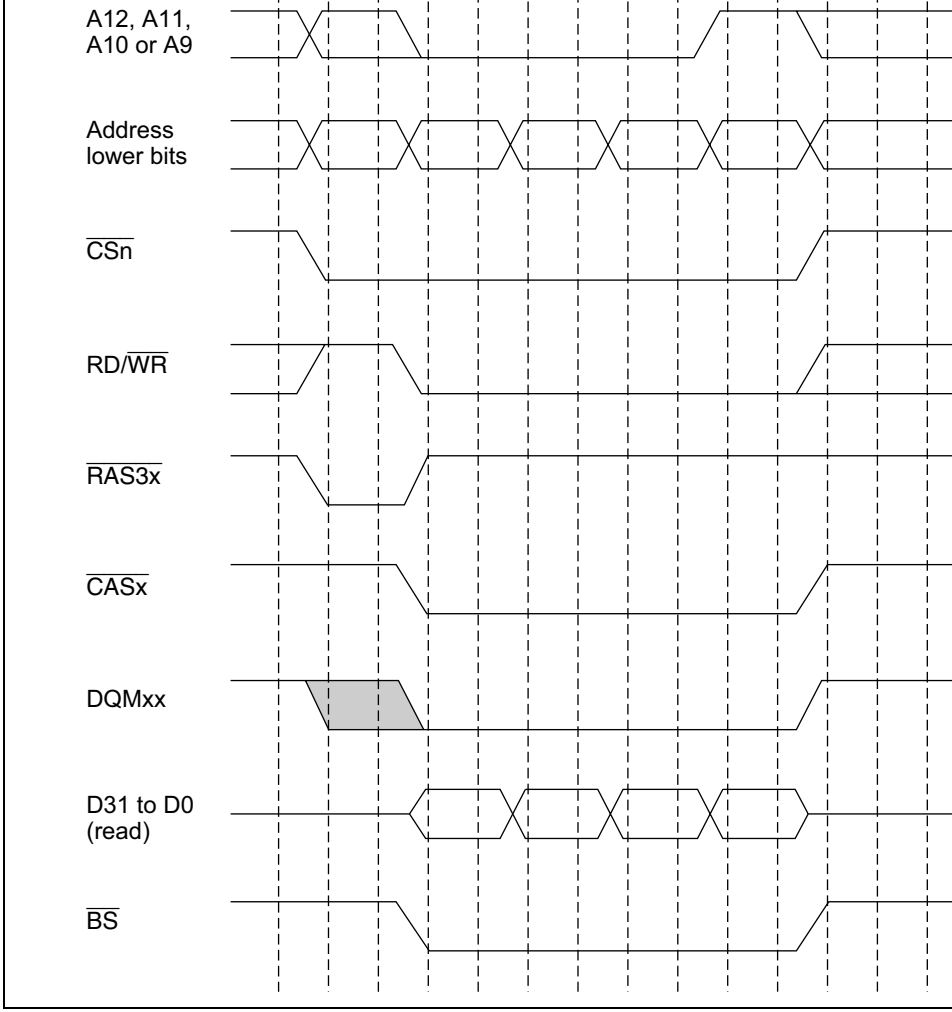


Figure 10.17 Basic Timing for Synchronous DRAM Burst Write

The number of Trwl cycles can be specified by the TRWL bits in MCR.

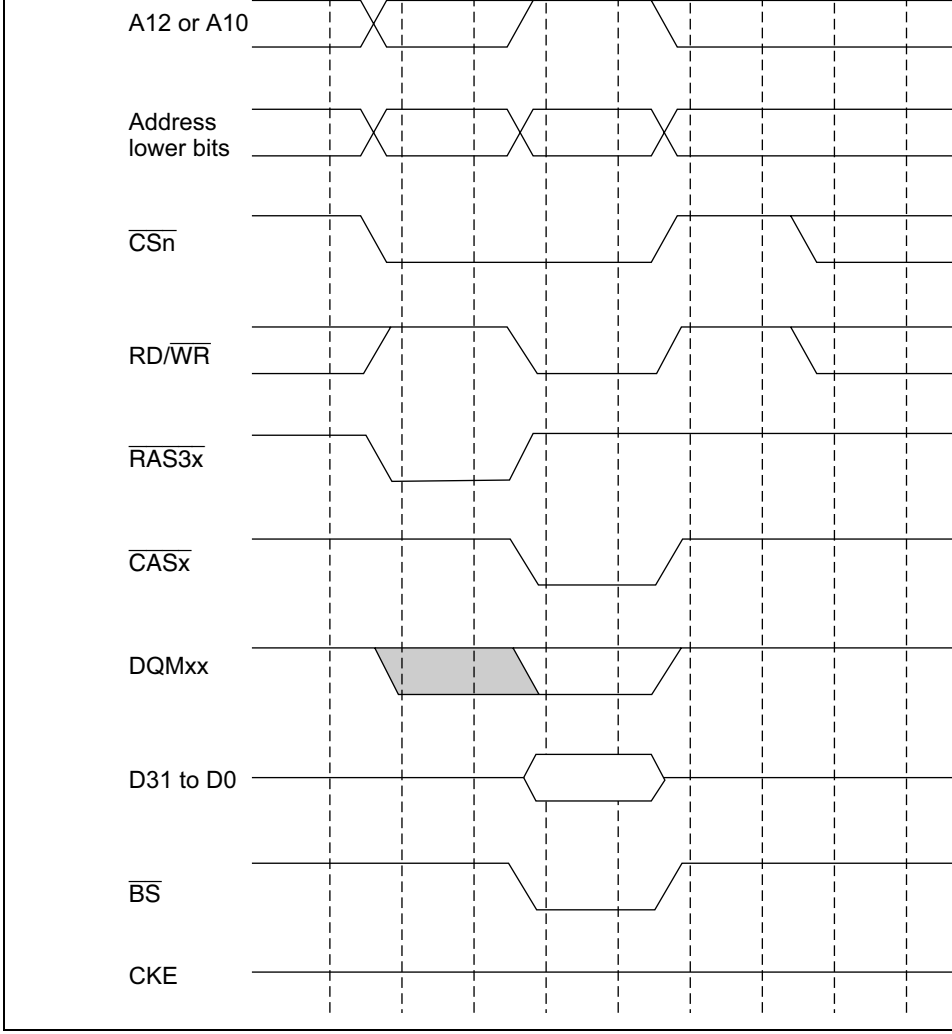


Figure 10.18 Basic Timing for Synchronous DRAM Single Write

command followed by a READ or WRIT command. If this is followed by an access to row address, the access time will be longer because of the precharging performed after request is issued.

In a write, when auto-precharge is performed, a command cannot be issued for a period Tpc cycles after issuance of the WRITA command. When bank active mode is used, R WRIT commands can be issued successively if the row address is the same. The numb can thus be reduced by Trwl + Tpc cycles for each write. The number of cycles betwe of the precharge command and the row address strobe command is determined by the MCR.

Whether faster execution speed is achieved by use of bank active mode or by use of ba determined by the probability of accessing the same row address (P1), and the average cycles from completion of one access to the next access (Ta). If Ta is greater than Tpc due to the precharge wait when writing is imperceptible. In this case, the access speed active mode and basic access is determined by the number of cycles from the start of a issuance of the read/write command: $(Tpc + Trcd) \times (1 - P1)$ and Trcd, respectively.

There is a limit on Tras, the time for placing each bank in the active state. If there is no that there will not be a cache hit and another row address will be accessed within the p which this value is maintained by program execution, it is necessary to set auto-refresh refresh cycle to no more than the maximum value of Tras. In this way, it is possible to restrictions on the maximum active state time for each bank. If auto-refresh is not used must be taken in the program to ensure that the banks do not remain active for longer t prescribed time.

A burst read cycle without auto-precharge is shown in figure 10.19, a burst read cycle row address in figure 10.20, and a burst read cycle for different row addresses in figur Similarly, a burst write cycle without auto-precharge is shown in figure 10.22, a burst for the same row address in figure 10.23, and a burst write cycle for different row add figure 10.24.

considered, as long as accesses to the same row address continue, the operation starts w
cycle in figure 10.19 or 10.22, followed by repetition of the cycle in figure 10.20 or 10.
access to a different area 3 space during this time has no effect. If there is an access to a
row address in the bank active state, after this is detected the bus cycle in figure 10.21 c
executed instead of that in figure 10.20 or 10.23. In bank active mode, too, all banks be
inactive after a refresh cycle or after the bus is released as the result of bus arbitration.

The bank active mode should not be used unless the bus width for all areas is 32 bits.

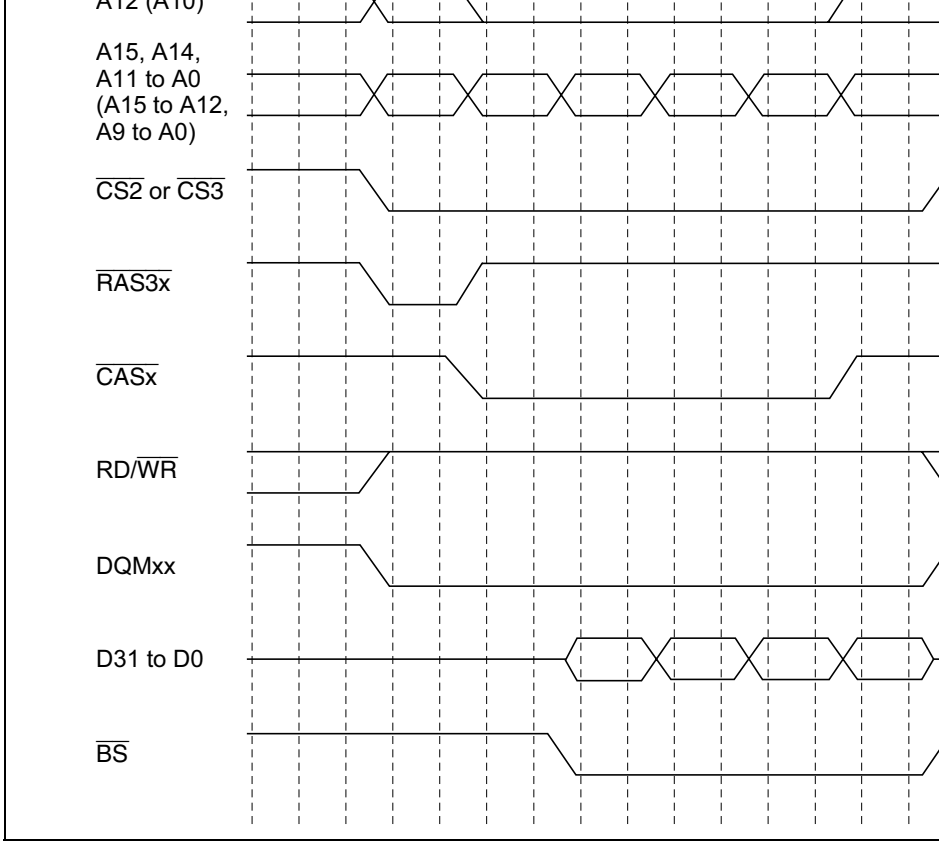


Figure 10.19 Burst Read Timing (No Precharge)

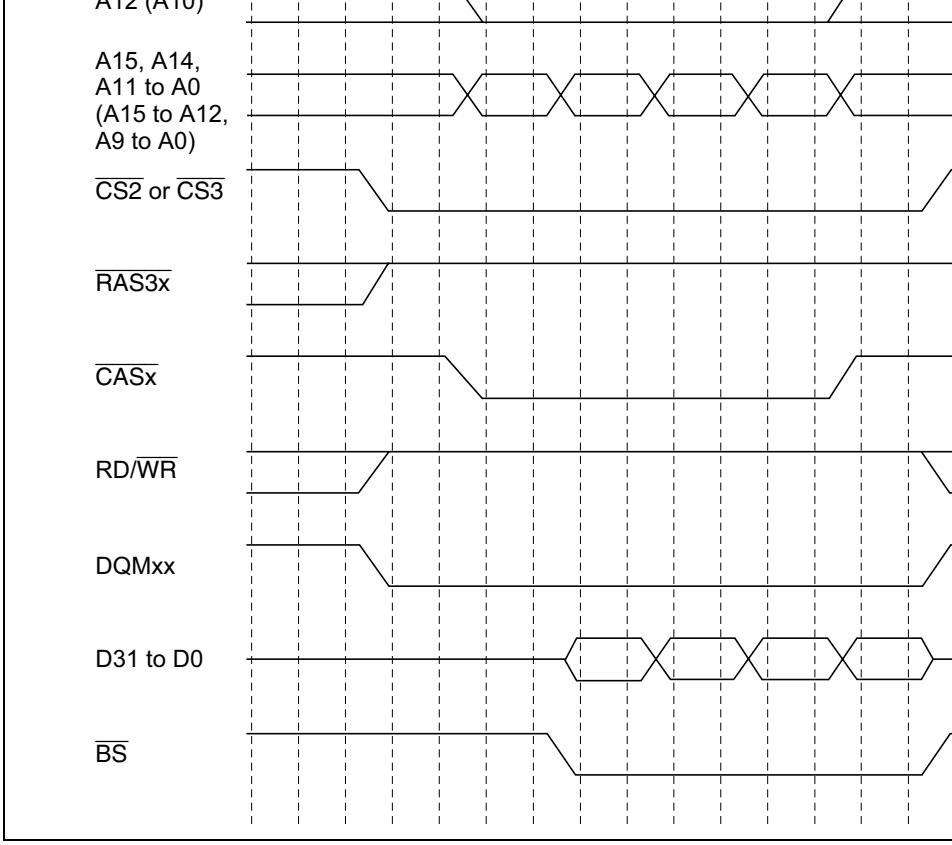


Figure 10.20 Burst Read Timing (Same Row Address)

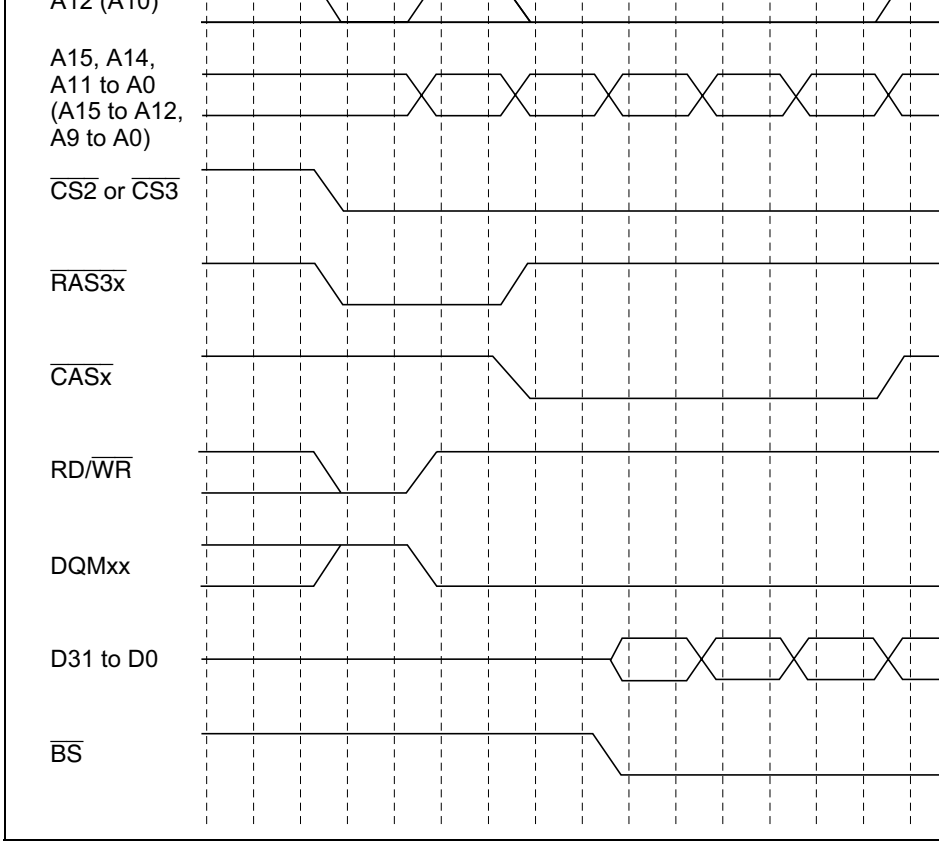


Figure 10.21 Burst Read Timing (Different Row Addresses)

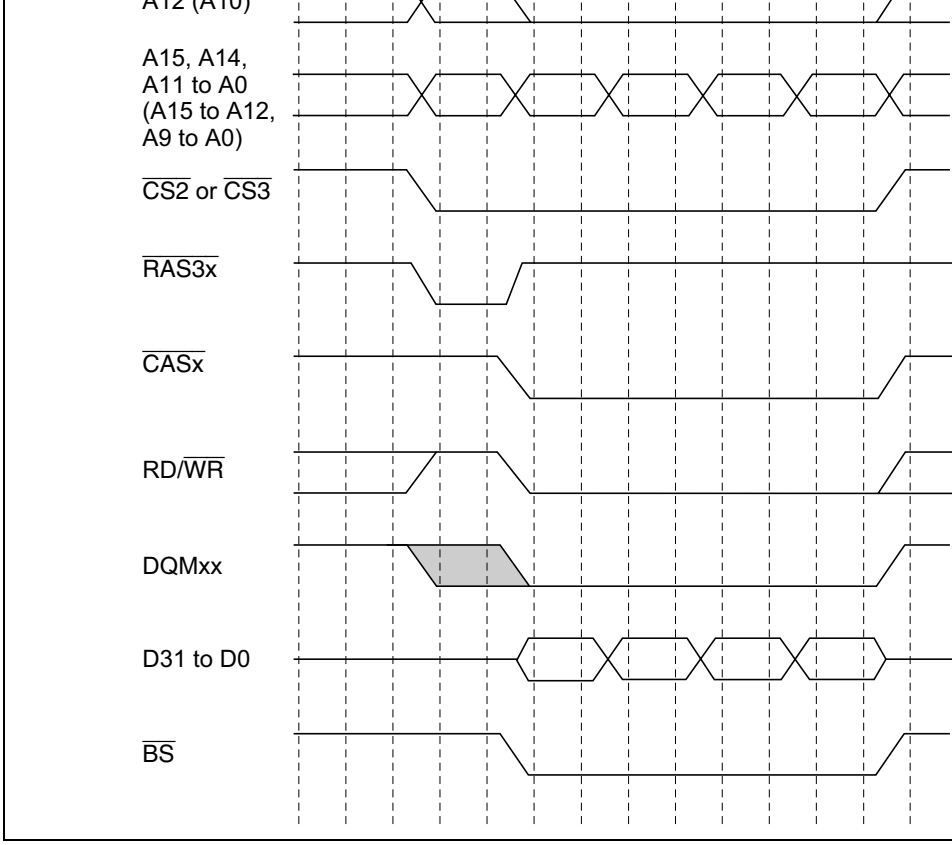


Figure 10.22 Burst Write Timing (No Precharge)

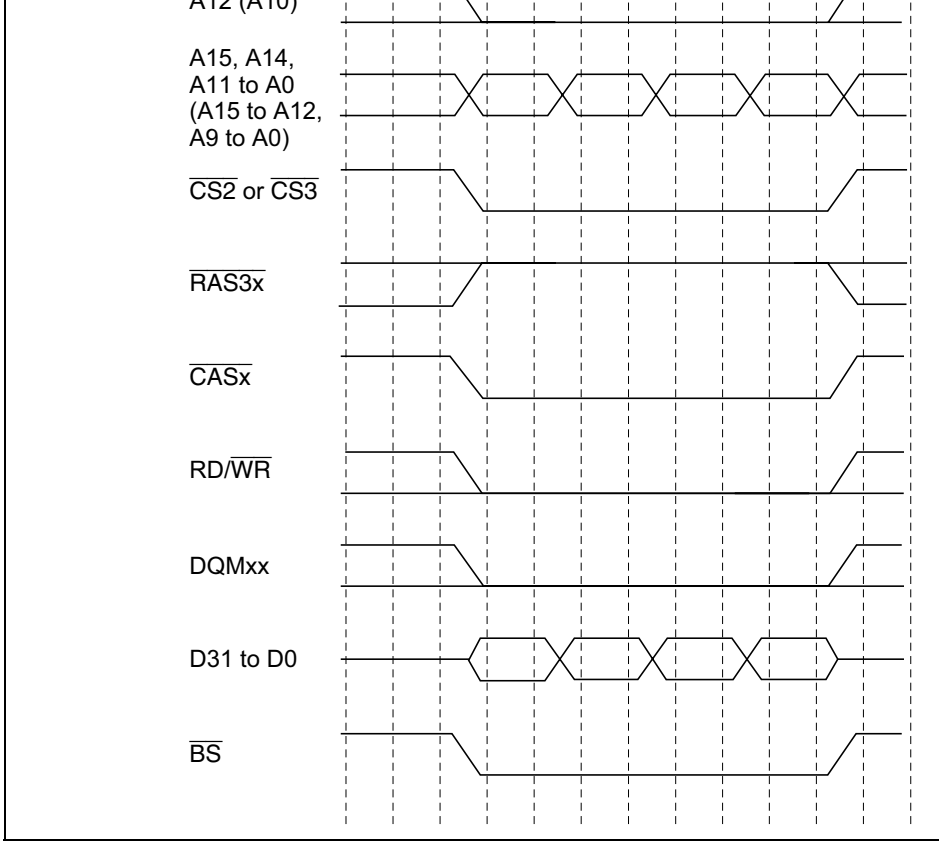


Figure 10.23 Burst Write Timing (Same Row Address)

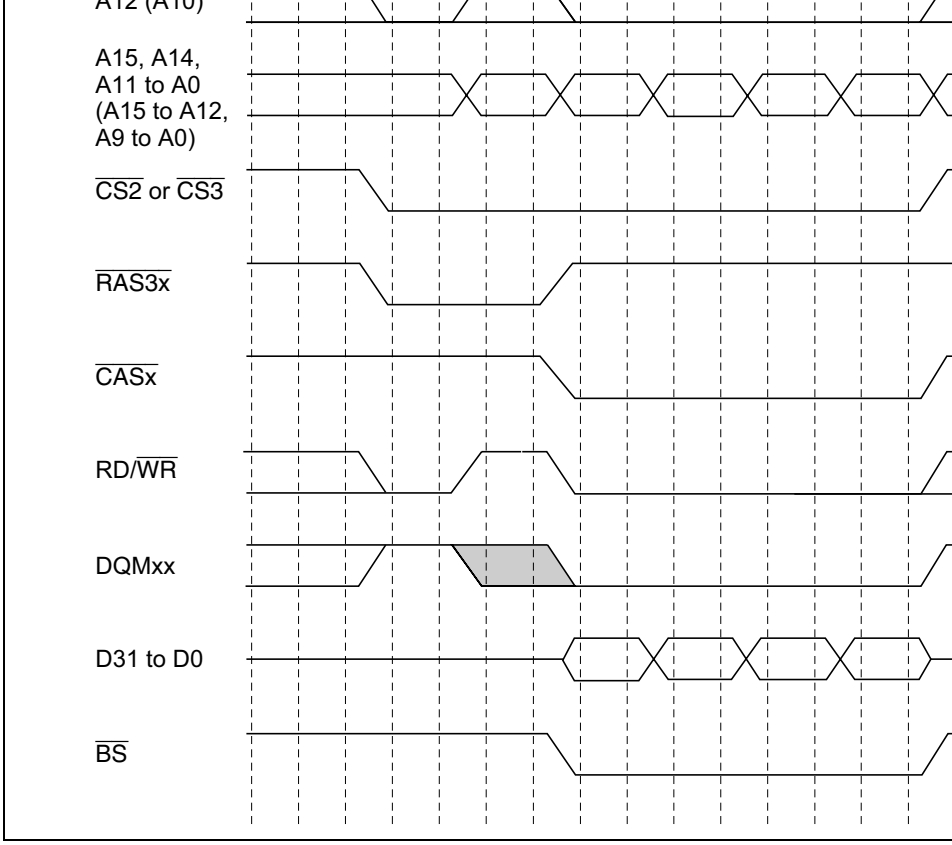


Figure 10.24 Burst Write Timing (Different Row Addresses)

RTCSR, and the value set in RTCOR. The value of bits CKS2-0 in RTCOR should to satisfy the refresh interval stipulation for the synchronous DRAM used. First ma settings for RTCOR, RTCNT, and the RMODE and RFSH bits in MCR, then mak CKS0 setting. When the clock is selected by CKS2-CKS0, RTCNT starts counting value at that time. The RTCNT value is constantly compared with the RTCOR val two values are the same, a refresh request is generated and an auto-refresh is perfo same time, RTCNT is cleared to zero and the count-up is restarted. Figure 10.25 sh auto-refresh cycle timing.

All-bank precharging is performed in the Tp cycle, then an REF command is issue cycle following the interval specified by the TPC bits in MCR. After the TRr cycle command output cannot be performed for the duration of the number of cycles spe TRAS bits in MCR plus the number of cycles specified by the TPC bits in MCR. T and TPC bits must be set so as to satisfy the synchronous DRAM refresh cycle tim (active/active command delay time).

Auto-refreshing is performed in normal operation, in sleep mode, and in case of a reset.

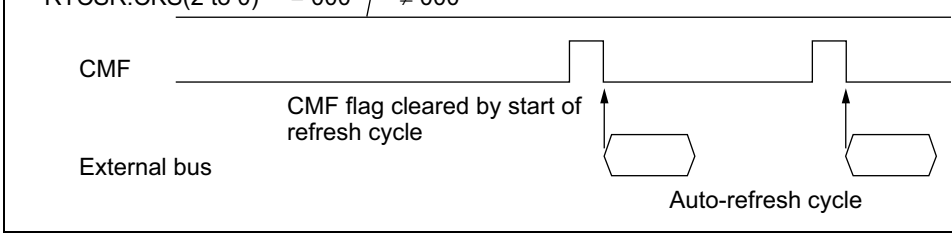


Figure 10.25 Auto-Refresh Operation

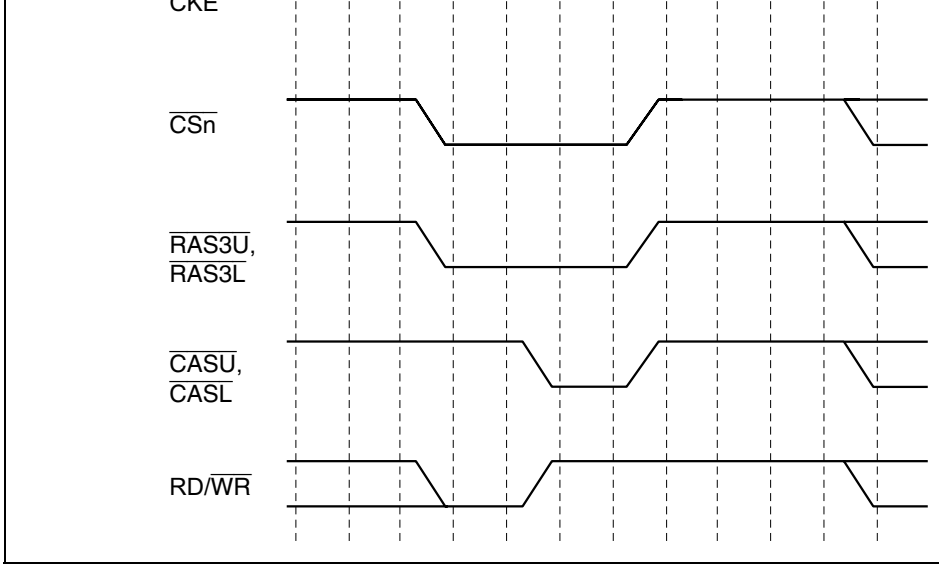


Figure 10.26 Synchronous DRAM Auto-Refresh Timing

clearing and data retention are performed correctly, and auto-refreshing is performe
correct intervals. When self-refreshing is activated from the state in which auto-refr
set, or when exiting standby mode other than through a power-on reset, auto-refresh
restarted if RFSH is set to 1 and RMODE is cleared to 0 when self-refresh mode is
the transition from clearing of self-refresh mode to the start of auto-refreshing takes
time should be taken into consideration when setting the initial value of RTCNT. M
RTCNT value 1 less than the RTCOR value will enable refreshing to be started imm

After self-refreshing has been set, the self-refresh state continues even if the chip st
is entered using the SH7709S's standby function, and is maintained even after recov
standby mode other than through a power-on reset. In case of a power-on reset, the
controller's registers are initialized, and therefore the self-refresh state is cleared.

Self-refreshing is performed in normal operation, in sleep mode, in standby mode, a
of a manual reset.

When using synchronous DRAM, use the following procedure to initiate self-refres

1. Clear the refresh control bit to 0.
2. Write H'00 to the RTCNT register.
3. Set the refresh control bit and refresh mode bit to 1.

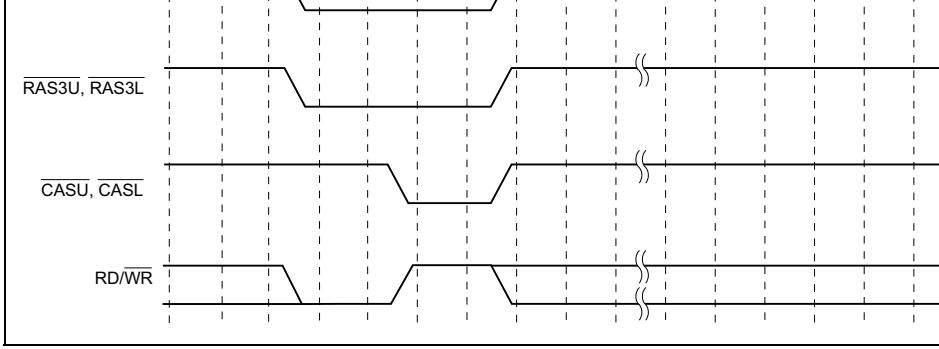


Figure 10.27 Synchronous DRAM Self-Refresh Timing

- Relationship between Refresh Requests and Bus Cycle Requests

  If a refresh request is generated during execution of a bus cycle, execution of the re
  deferred until the bus cycle is completed. If a refresh request occurs when the bus
  released by the bus arbiter, refresh execution is deferred until the bus is acquired. I
  between RTCNT and RTCOR occurs while a refresh is waiting to be executed, so
  refresh request is generated, the previous refresh request is eliminated. In order for
  to be performed normally, care must be taken to ensure that no bus cycle or bus rig
  that is longer than the refresh interval. When a refresh request is generated, the $\overline{\text{IRQ}}$
  asserted (driven low). Therefore, normal refreshing can be performed by having th
  pin monitored by a bus master other than the SH7709S requesting the bus, or the b
  and returning the bus to the SH7709S. When refreshing is started, and if no other i
  request has been generated, the $\overline{\text{IRQOUT}}$ pin is negated (driven high).

wrap type = sequential, and burst length 1 supported by the SH7709S, arbitrary data is
byte-size access to the following addresses.

With 32-bit bus width:

	Area 2	Area 3
CAS latency 1	FFFFD840	FFFFE840
CAS latency 2	FFFFD880	FFFFE880
CAS latency 3	FFFFD8C0	FFFFE8C0

With 16-bit bus width:

	Area 2	Area 3
CAS latency 1	FFFFD420	FFFFE420
CAS latency 2	FFFFD440	FFFFE440
CAS latency 3	FFFFD460	FFFFE460

Mode register setting timing is shown in figure 10.28.

As a result of the write to address H'FFFFD000 + X or H'FFFFE000 + X, a precharge a
(PALL) command is first issued in the TRp1 cycle, then a mode register write comman
in the TMw1 cycle.

Address signals, when the mode-register write command is issued, are as follows:

32-bit bus width:
- A15–A9 = 0000100 (burst read and single write)
- A8–A6 = CAS latency
- A5 = 0 (burst type = sequential)
- A4–A2 = 000 (burst length 1)

16-bit bus width:
- A14–A8 = 0000100 (burst read and single write)
- A7–A5 = CAS latency
- A4 = 0 (burst type = sequential)
- A3–A1 = 000 (burst length 1)

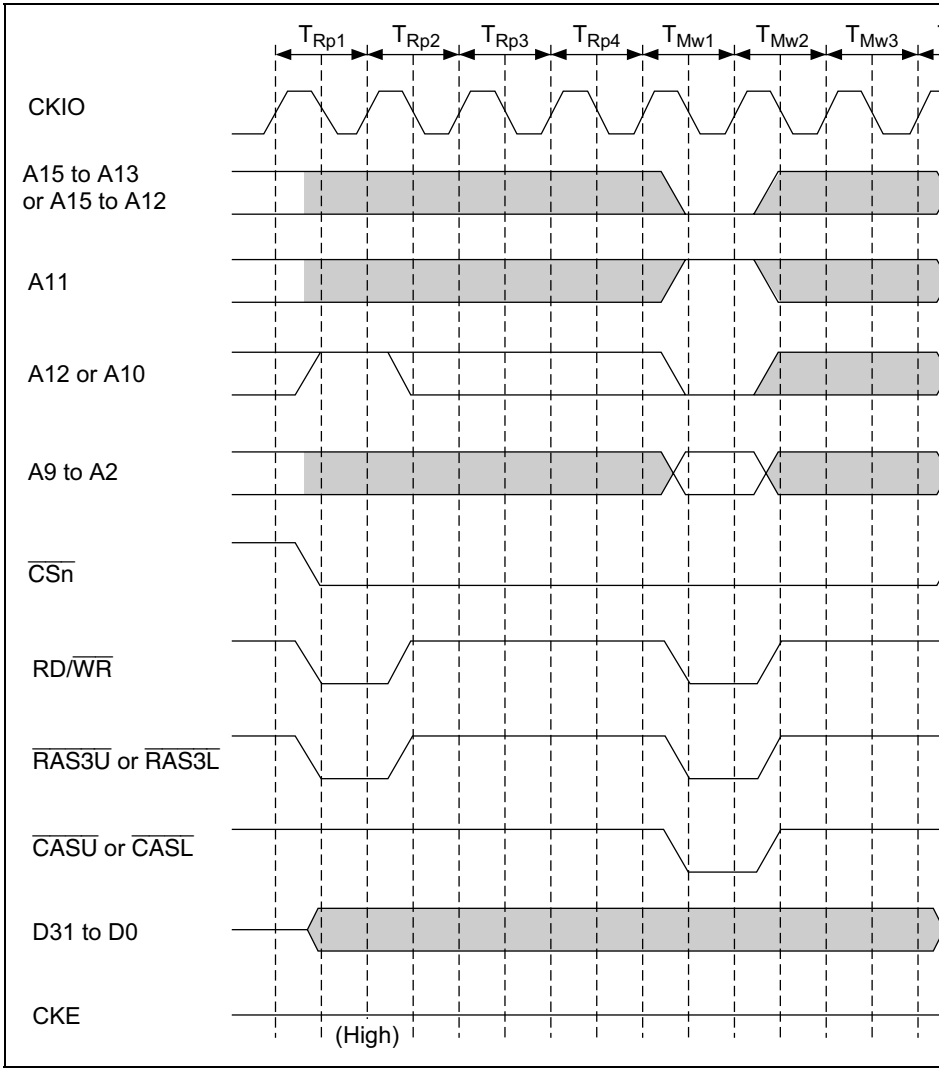


Figure 10.28 Synchronous DRAM Mode Write Timing

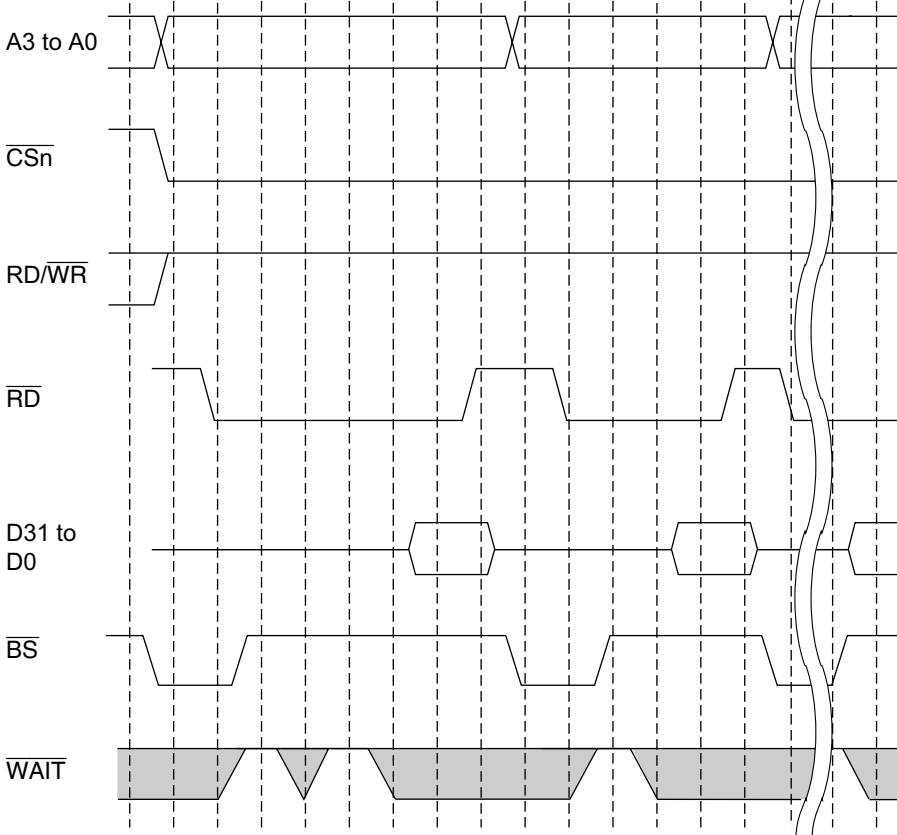
When 16-bit ROM is connected, 4 or 8 can be set in the same way. When 32-bit ROM i
connected, only 4 can be set.

$\overline{\text{WAIT}}$ pin sampling is performed in the first access if one or more wait states are set, an
always performed in the second and subsequent accesses.

The second and subsequent access cycles also comprise two cycles when a burst ROM
made and the wait specification is 0. The timing in this case is shown in figure 10.30.

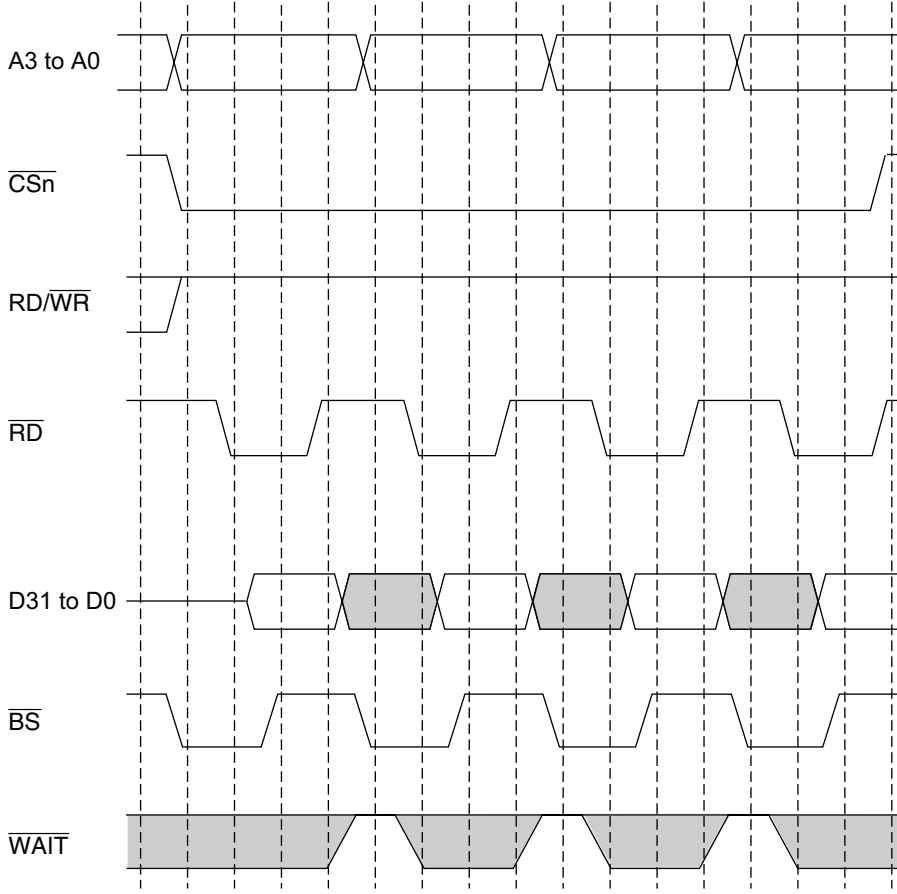
However, the $\overline{\text{WAIT}}$ signal is ignored in the following three cases:

- A write to external address space in dual address mode with 16-byte DMA transfer
- Transfer from an external device with DACK to external address space in single ad
  with 16-byte DMA transfer
- Cache write-back access



Note: For a write cycle, a basic bus cycle (write cycle) is performed.

Figure 10.29 Burst ROM Wait Access Timing



Note: For a write cycle, a basic bus cycle (write cycle) is performed.

Figure 10.30 Burst ROM Basic Access Timing

Figure 10.31 shows an example of PCMCIA card connection to the SH7709S. To ena
insertion of the PCMCIA cards (i.e. insertion or removal while system power is being
3-state buffer must be connected between the SH7709S's bus interface and the PCMC

As operation in big-endian mode is not explicitly stipulated in the JEIDA/PCMCIA sp
the PCMCIA interface for the SH7709S in big-endian mode is stipulated independentl

However, the $\overline{\text{WAIT}}$ signal is ignored in the following three cases:

- A write to external address space in dual address mode with 16-byte DMA transfe
- Transfer from an external device with DACK to external address space in single a
  with 16-byte DMA transfer
- Cache write-back access

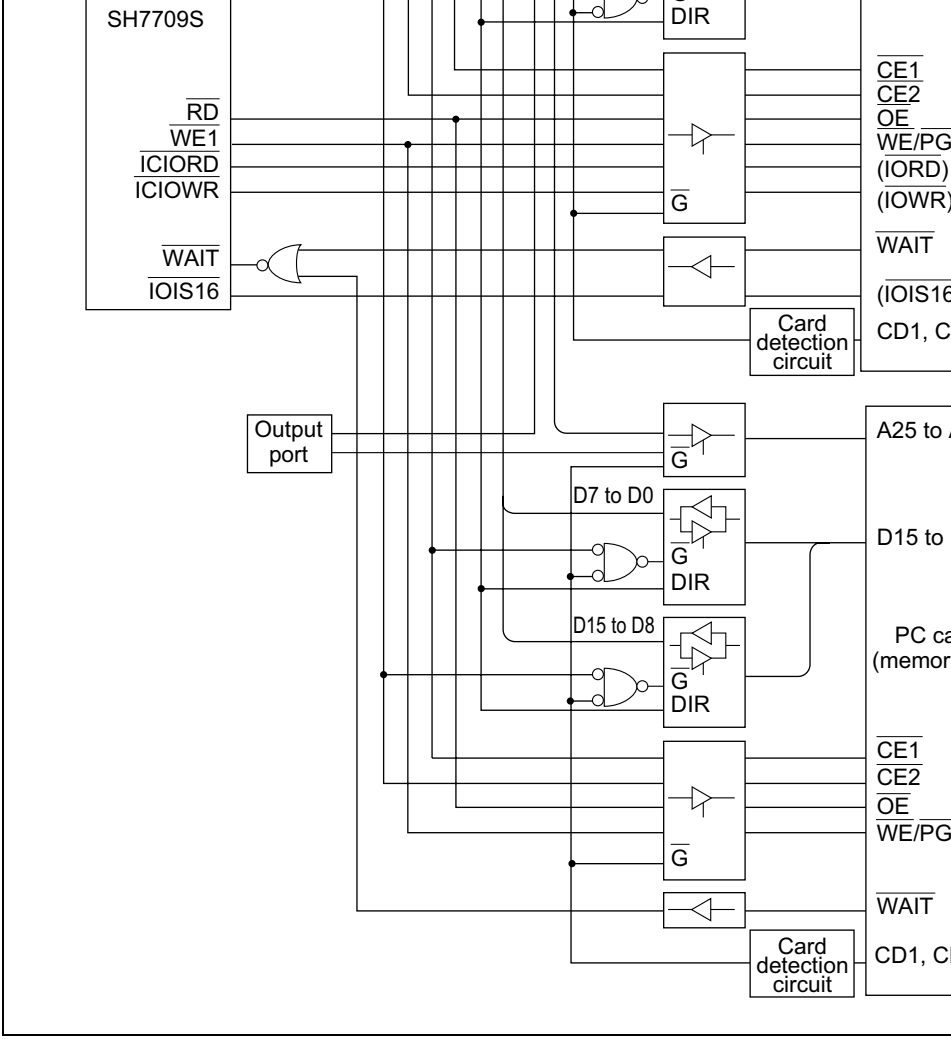


Figure 10.31 Example of PCMCIA Interface

WAIT pin can be inserted in the same way as for the basic interface. Figure 10.33 sho
PCMCIA memory bus wait timing.

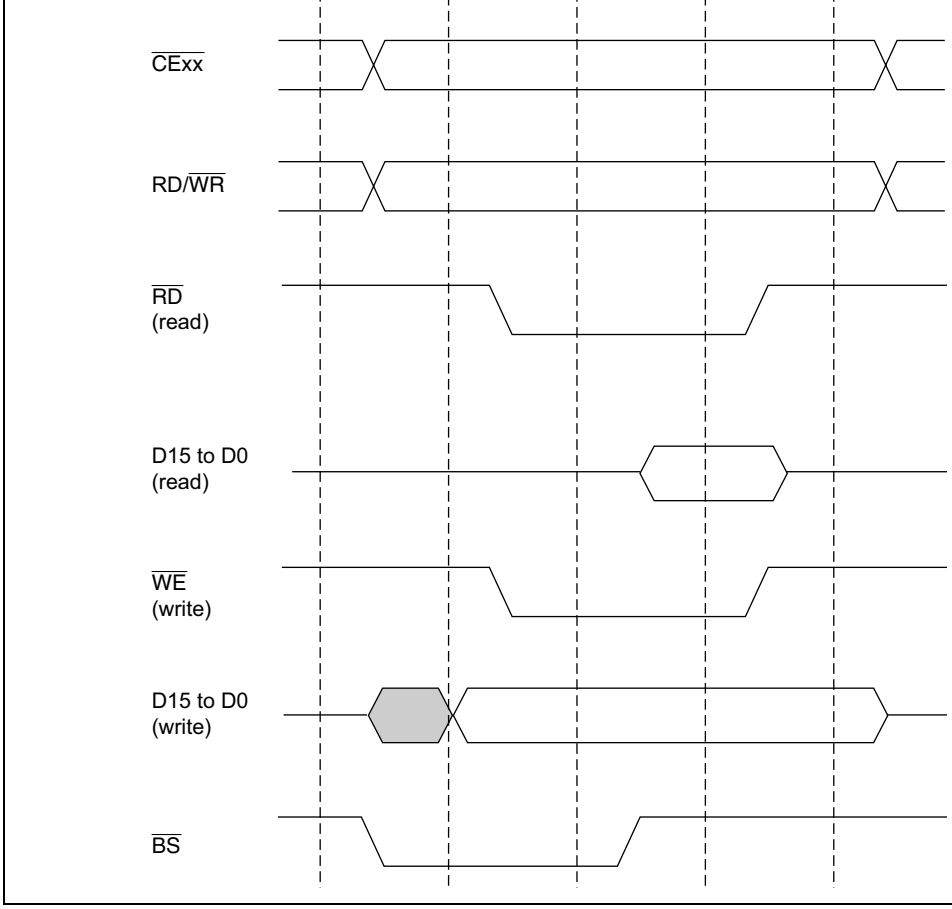


Figure 10.32 Basic Timing for PCMCIA Memory Card Interface

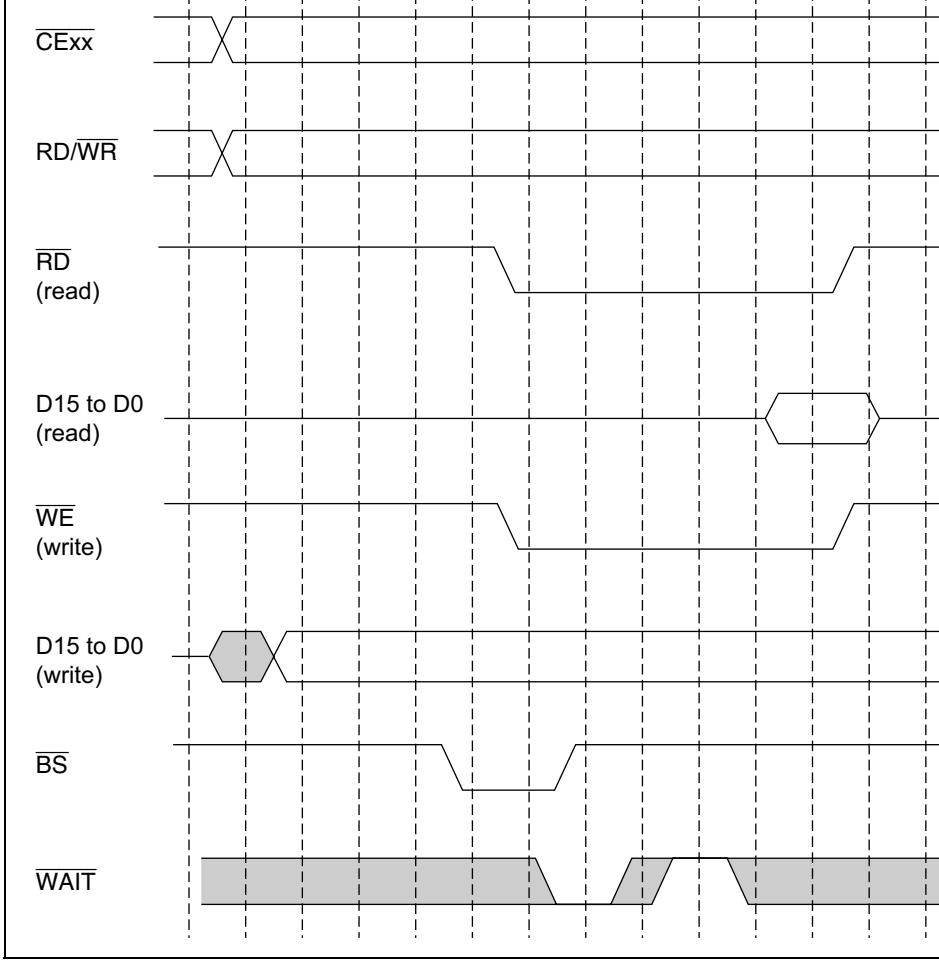


Figure 10.33 Wait Timing for PCMCIA Memory Card Interface

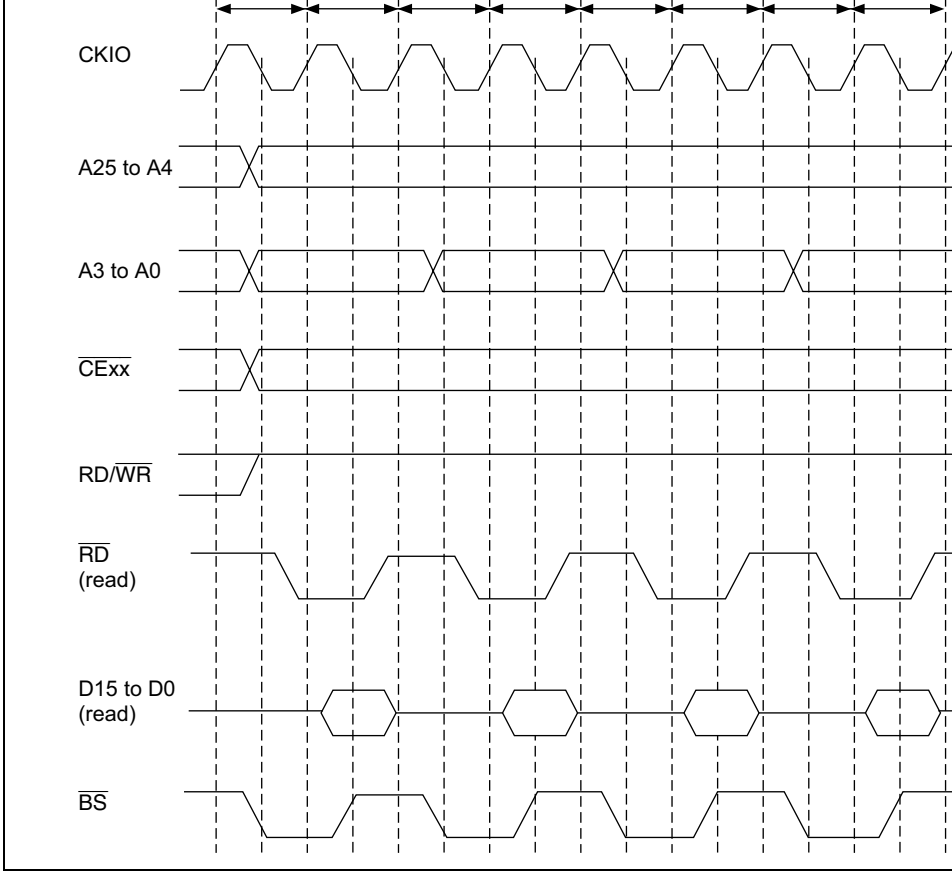


Figure 10.34 Basic Timing for PCMCIA Memory Card Interface Burst Ac

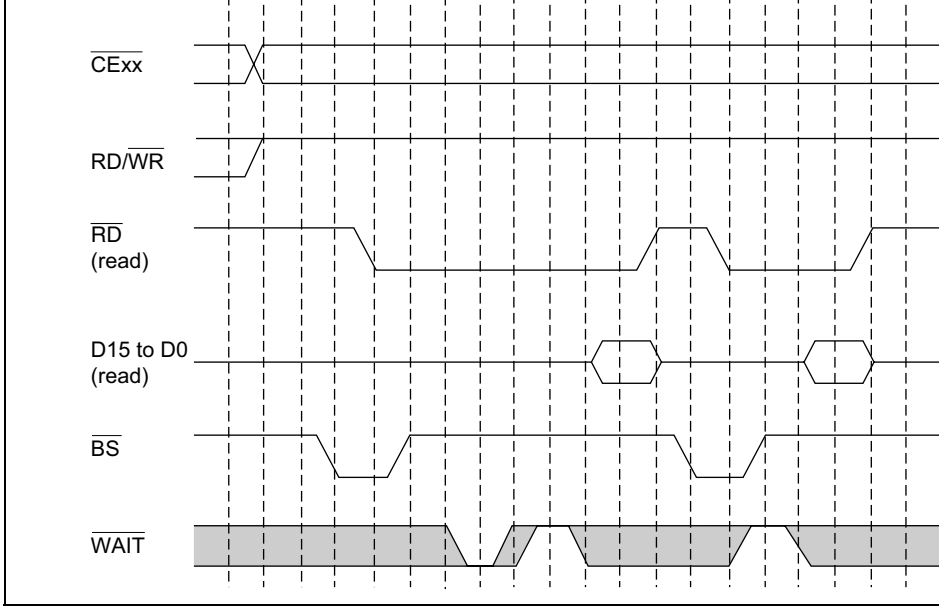


Figure 10.35 Wait Timing for PCMCIA Memory Card Interface Burst A

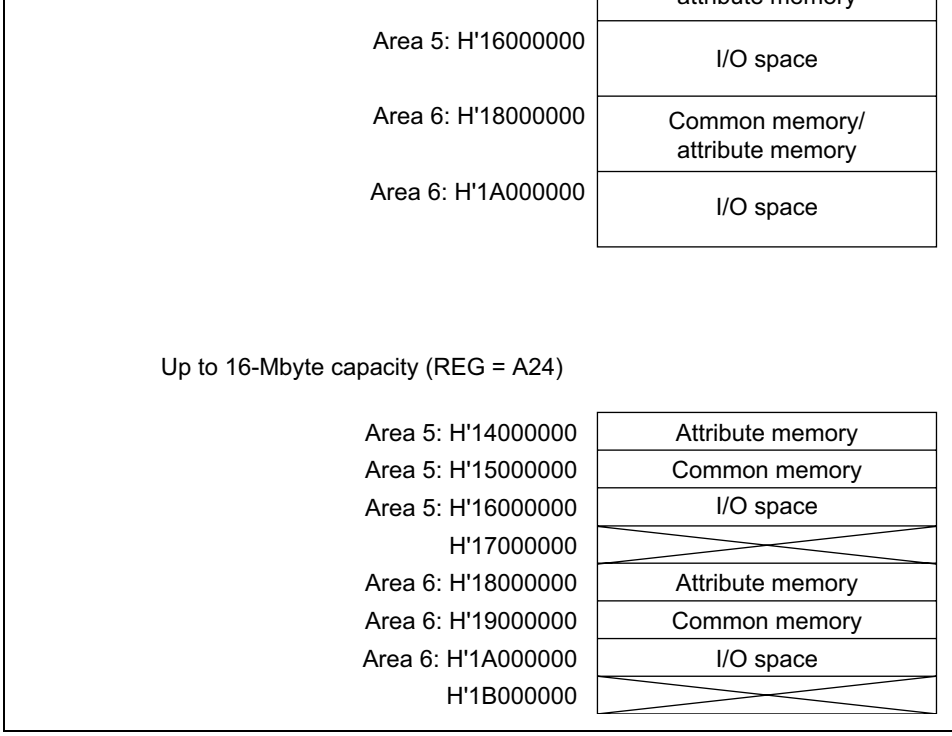


Figure 10.36 PCMCIA Space Allocation

When accessing a PCMCIA I/O card, the access should be performed using a non-cac
in virtual space (P2 or P3 space) or an area specified as non-cacheable by the MMU.

When an I/O card interface access is made to a PCMCIA card in little-endian mode, d
sizing of the I/O bus width is possible using the $\overline{\text{IOIS16}}$ pin. When a 16-bit bus width
5 or area 6, if the $\overline{\text{IOIS16}}$ signal is high during a word-size I/O bus cycle, the I/O port
recognized as being 8 bits in width. In this case, a data access for only 8 bits is perforn
I/O bus cycle being executed, followed automatically by a data access for the remainir

Figure 10.39 shows the basic timing for dynamic bus sizing.

In big-endian mode, the $\overline{\text{IOIS16}}$ signal is not supported, and should be fixed low.

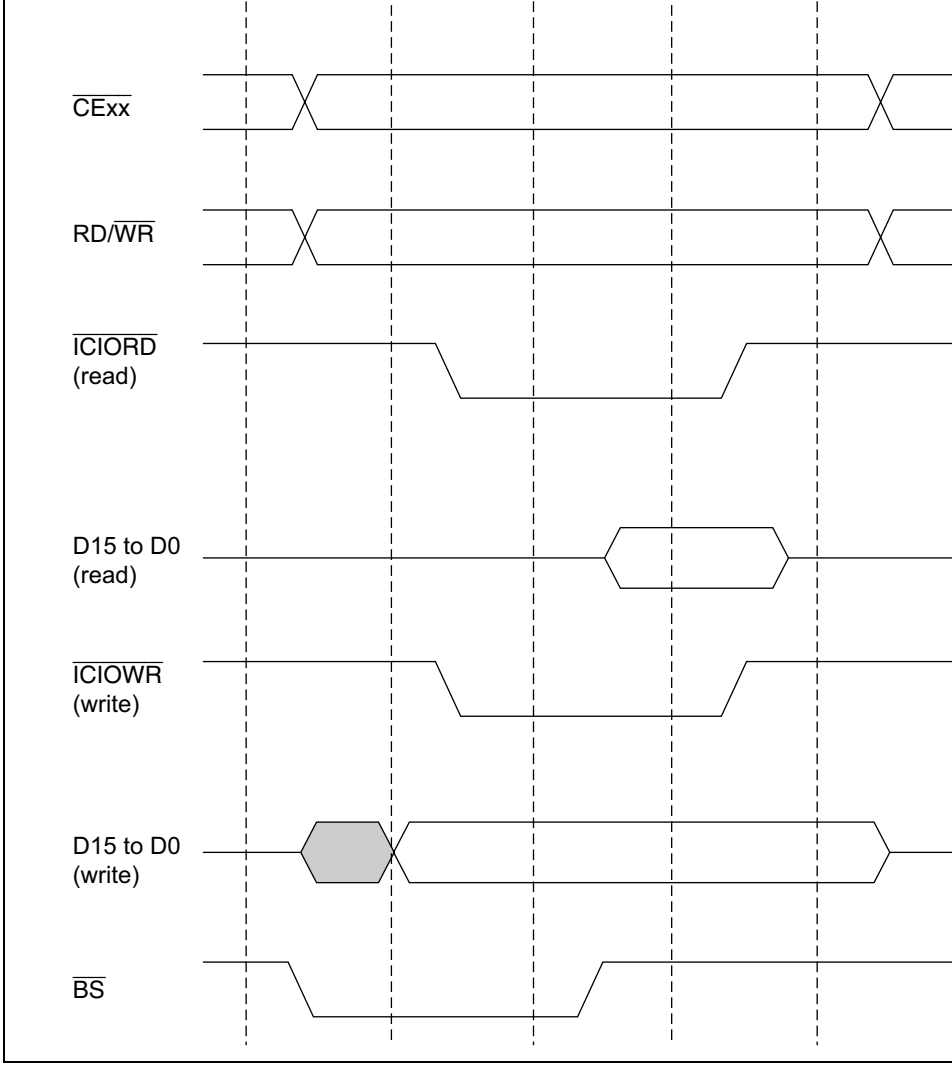


Figure 10.37 Basic Timing for PCMCIA I/O Card Interface

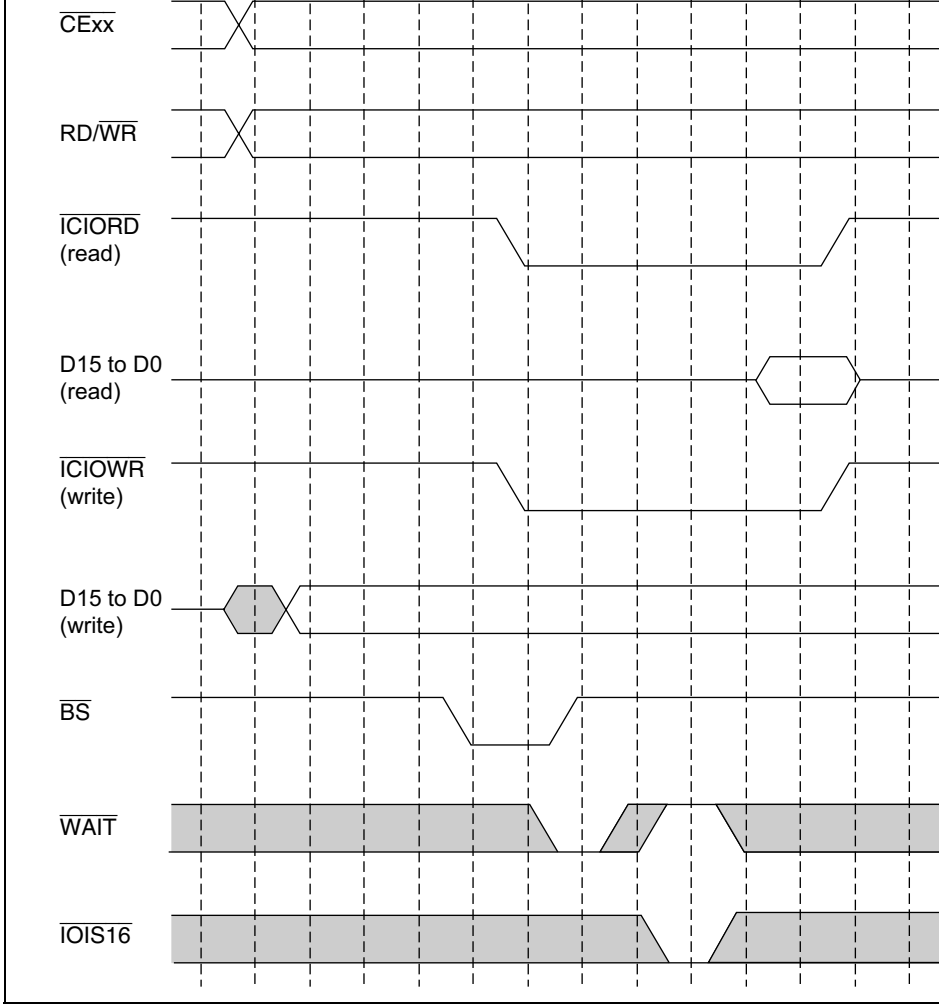


Figure 10.38 Wait Timing for PCMCIA I/O Card Interface

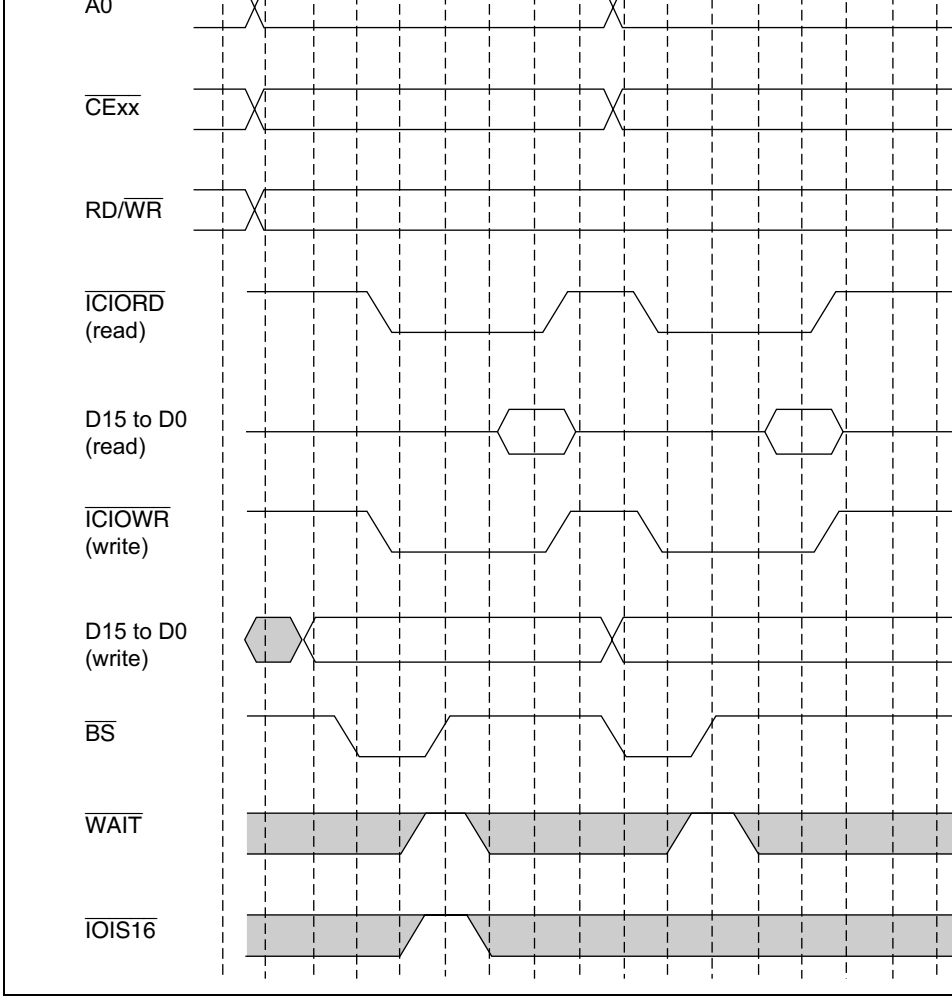


Figure 10.39 Dynamic Bus Sizing Timing for PCMCIA I/O Card Interfa

different area, and when a read access is followed by a write access from the SH7709S SH7709S performs consecutive write cycles, the data transfer direction is fixed (from SH7709S to other memory) and there is no problem. With read accesses to the same a principle, data is output from the same data buffer, and wait cycle insertion is not perf AnIW1 and AnIW0 (n = 0, 2–6) in WCR1 specify the number of idle cycles to be inse between access cycles when a physical space area access is followed by an access to a or when the SH7709S performs a write access after a read access to physical space are is originally space between accesses, the number of idle cycles inserted is the specifie idle cycles minus the number of empty cycles.

Waits are not inserted between accesses when bus arbitration is performed, since empt inserted for arbitration purposes.

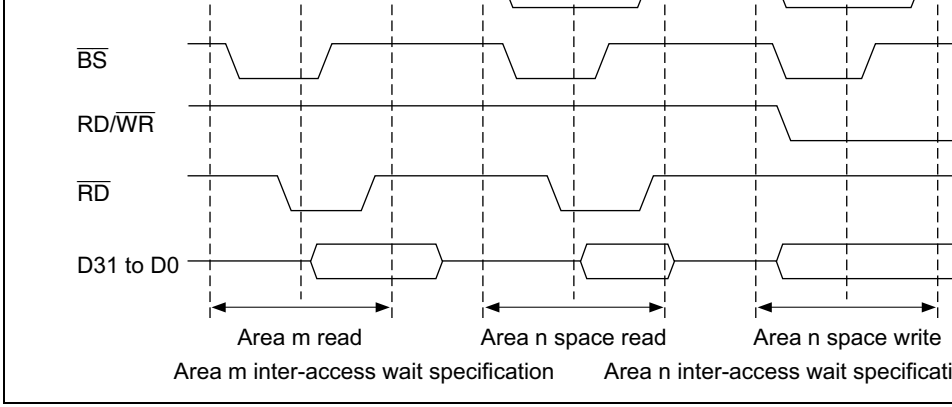


Figure 10.40 Waits between Access Cycles

10.3.8 Bus Arbitration

When a bus release request ($\overline{\text{BREQ}}$) is asserted from an external device, buses are relea
the bus cycle being executed is completed and a bus grant signal ($\overline{\text{BACK}}$) is output. Th
released during burst transfers for cache fills or write-back, or TAS instruction executio
the read cycle and write cycle. Bus arbitration is not executed in multiple bus cycles tha
generated when the data bus width is shorter than the access size; i.e. in the bus cycles w
longword access is executed for the 8-bit memory. At the negation of $\overline{\text{BREQ}}$, $\overline{\text{BACK}}$ is
and bus use is restarted. See Appendix A.1, Pin States, for the pin states when the bus is

The SH7709S sometimes needs to retrieve a bus it has released. For example, when me
generates a refresh request or an interrupt request internally, the SH7709S must perform
appropriate processing. The SH7709S has a bus request signal ($\overline{\text{IRQOUT}}$) for this purp
it must retrieve the bus, it asserts the $\overline{\text{IRQOUT}}$ signal. Devices asserting an external bus
request receive the assertion of the $\overline{\text{IRQOUT}}$ signal and negate the $\overline{\text{BREQ}}$ signal to rele
The SH7709S retrieves the bus and carries out the processing.

PULA bit in BCR1 to 1. The address pins are pulled up for a 4-clock period after $\overline{\text{BAC}}$ asserted. Figure 10.41 shows the address pin pull-up timing. Similarly, data pin pull-u performed by setting the PULD bit in BCR1 to 1. The data pins should be pulled up w bus is not in use. The data pin pull-up timing for a read cycle is shown in figure 10.42 timing for a write cycle in figure 10.43.

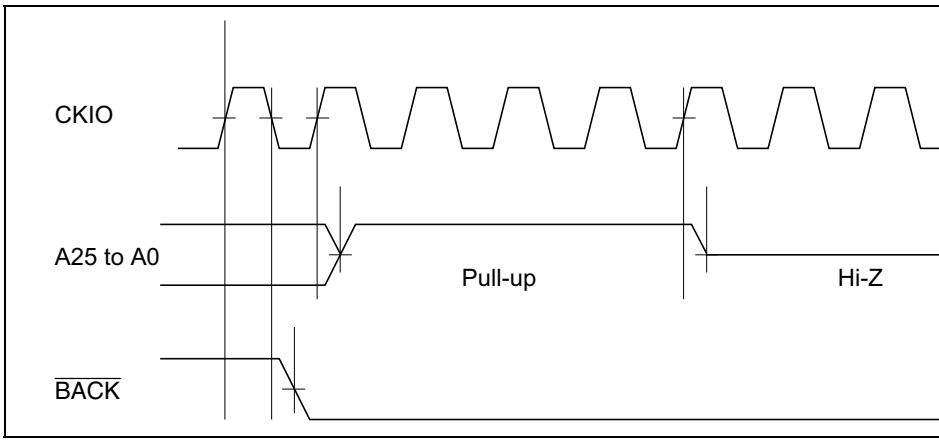


Figure 10.41 Pull-Up Timing for Pins A25 to A0

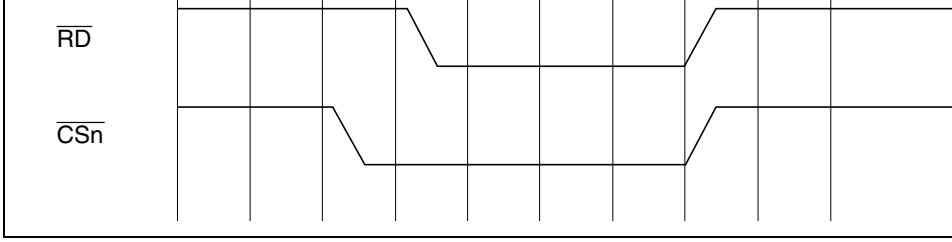


Figure 10.42 Pull-Up Timing for Pins D31 to D0 (Read Cycle)

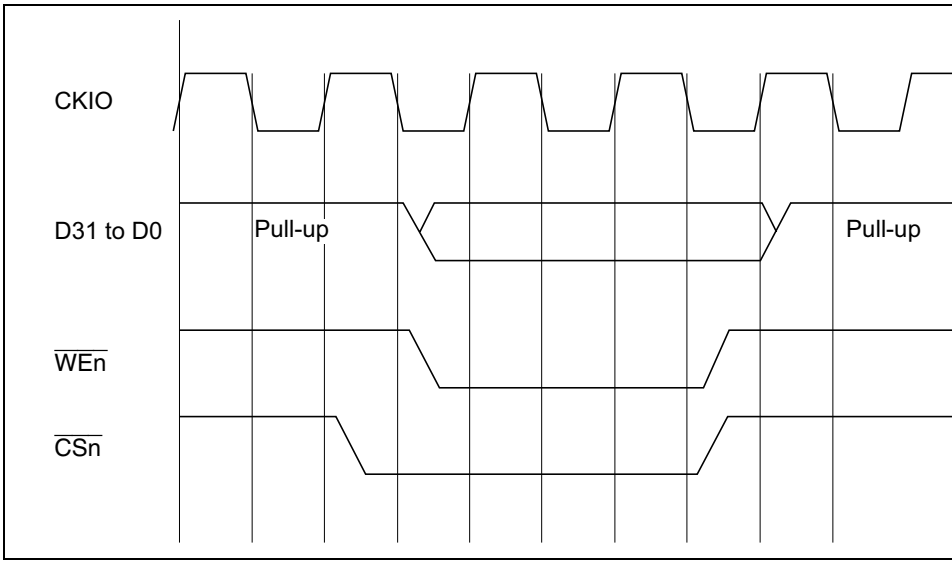


Figure 10.43 Pull-Up Timing for Pins D31 to D0 (Write Cycle)

MCS[0]–MCS[7], the corresponding bits in the PCCR register should be set to "other

When CS2/0 = 0 in the MCSCR0 and when the PTC0 pin is switched to $\overline{\text{MCS[0]}}$ (whe
PCOMD1–PCOMD0 are set to “other function”), the $\overline{\text{CS0}}$ pin is also switched to $\overline{\text{MCS}}$

As port register writes operate on the peripheral clock, they take time compared with i
execution by the CPU operating on the high-speed internal clock. Therefore, if an inst
accesses $\overline{\text{MCS[1]}}$ to $\overline{\text{MCS[7]}}$ is located several instructions after an instruction that sw
to $\overline{\text{MCS}}$, the switch from PTC[n] to $\overline{\text{MCS}}$n and from $\overline{\text{CS0}}$ to $\overline{\text{MCS[0]}}$ may not be perfo
correctly.

To prevent this problem, the following switching procedure should be used.

- When the program runs with cache on

(1) To switch port C to $\overline{\text{MCS}}$, set the corresponding bits in the PCCR register to 00 ("
function").

(2) Read the PCCR register and check whether the set value is read. Repeat until the s
read.

(3) Perform a dummy read from non-cacheable CS0 space (e.g. address H'A0000000)
result in an access to the CS0 space, and immediately afterward, CS0 will be switc
$\overline{\text{MCS[0]}}$, and port C[n] will be switched to $\overline{\text{MCS[n]}}$.

(4) Access can now be made to the $\overline{\text{MCS[1]}}$ to $\overline{\text{MCS[7]}}$ spaces.

- When the program runs in $\overline{\text{MCS[0]}}$ space with cache off

(1) Set the PCCR register as in (1) above.

(2) Place at least three NOP instructions after the instruction in (1). As a result, when
register is rewritten, an access to the CS0 space will be generated, and immediatel
CS0 will be switched to $\overline{\text{MCS[0]}}$, and port C[n] will be switched to $\overline{\text{MCS[n]}}$.

(3) Access can now be made to the $\overline{\text{MCS[1]}}$ to $\overline{\text{MCS[7]}}$ spaces.

		1	1	—	—	L	H	H'3000000 to H'3FFFFFF	
0	1	0	0	0	—	L	H	H'0000000 to H'07FFFFF	64-
		0	0	1	—	L	H	H'0800000 to H'0FFFFFF	
		0	1	0	—	L	H	H'1000000 to H'17FFFFF	
		0	1	1	—	L	H	H'1800000 to H'1FFFFFF	
		1	0	0	—	L	H	H'2000000 to H'27FFFFF	
		1	0	1	—	L	H	H'2800000 to H'2FFFFFF	
		1	1	0	—	L	H	H'3000000 to H'37FFFFF	
		1	1	1	—	L	H	H'3800000 to H'3FFFFFF	
0	0	0	0	0	0	L	H	H'0000000 to H'03FFFFF	32-
		0	0	0	1	L	H	H'0400000 to H'07FFFFF	
		0	0	1	0	L	H	H'0800000 to H'0BFFFFF	
		0	0	1	1	L	H	H'0C00000 to H'0FFFFFF	
		0	1	0	0	L	H	H'1000000 to H'13FFFFF	
		0	1	0	1	L	H	H'1400000 to H'17FFFFF	
		0	1	1	0	L	H	H'1800000 to H'1BFFFFF	
		0	1	1	1	L	H	H'1C00000 to H'1FFFFFF	
		1	0	0	0	L	H	H'2000000 to H'23FFFFF	
		1	0	0	1	L	H	H'2400000 to H'27FFFFF	
		1	0	1	0	L	H	H'2800000 to H'2BFFFFF	
		1	0	1	1	L	H	H'2C00000 to H'2FFFFFF	
		1	1	0	0	L	H	H'3000000 to H'33FFFFF	
		1	1	0	1	L	H	H'3400000 to H'37FFFFF	
		1	1	1	0	L	H	H'3800000 to H'3BFFFFF	
		1	1	1	1	L	H	H'3C00000 to H'3FFFFFF	

0	1	0	0	0	—	H	L	H'0000000 to H'07FFFFF	6
		0	0	1	—	H	L	H'0800000 to H'0FFFFFF	
		0	1	0	—	H	L	H'1000000 to H'17FFFFF	
		0	1	1	—	H	L	H'1800000 to H'1FFFFFF	
		1	0	0	—	H	L	H'2000000 to H'27FFFFF	
		1	0	1	—	H	L	H'2800000 to H'2FFFFFF	
		1	1	0	—	H	L	H'3000000 to H'37FFFFF	
		1	1	1	—	H	L	H'3800000 to H'3FFFFFF	
0	0	0	0	0	0	H	L	H'0000000 to H'03FFFFF	3
		0	0	0	1	H	L	H'0400000 to H'07FFFFF	
		0	0	1	0	H	L	H'0800000 to H'0BFFFFF	
		0	0	1	1	H	L	H'0C00000 to H'0FFFFFF	
		0	1	0	0	H	L	H'1000000 to H'13FFFFF	
		0	1	0	1	H	L	H'1400000 to H'17FFFFF	
		0	1	1	0	H	L	H'1800000 to H'1BFFFFF	
		0	1	1	1	H	L	H'1C00000 to H'1FFFFFF	
		1	0	0	0	H	L	H'2000000 to H'23FFFFF	
		1	0	0	1	H	L	H'2400000 to H'27FFFFF	
		1	0	1	0	H	L	H'2800000 to H'2BFFFFF	
		1	0	1	1	H	L	H'2C00000 to H'2FFFFFF	
		1	1	0	0	H	L	H'3000000 to H'33FFFFF	
		1	1	0	1	H	L	H'3400000 to H'37FFFFF	
		1	1	1	0	H	L	H'3800000 to H'3BFFFFF	
		1	1	1	1	H	L	H'3C00000 to H'3FFFFFF	

the DMAC reduces the burden on the CPU and increases overall operating efficiency.

11.1.1 Features

The DMAC has the following features.

- Four channels
- 4-GB address space in the architecture
- 16-byte transfer (In 16-byte transfer, four 32-bit reads are executed, followed by fo writes.)
- Choice of 8-bit, 16-bit, 32-bit, or 16-byte transfer data length
- 16 Mbytes (16,777,216 transfers)
- Address mode: Dual address mode and single address mode are supported. In add address transfer mode or indirect address transfer mode can be selected.
  - Dual address mode transfer: Both the transfer source and transfer destination a by address. Dual address mode has direct address transfer mode and indirect ac transfer mode.

    Direct address transfer mode: The values specified in the DMAC registers ind transfer source and transfer destination. Two bus cycles are required for one d

    Indirect address transfer mode: Data is transferred with the address stored prio address specified in the transfer source address in the DMAC. Other operation same as those of direct address transfer mode. This function is only available i Four bus cycles are required for one data transfer.
  - Single address mode transfer: Either the transfer source or transfer destination device is accessed (selected) by means of the DACK signal, and the other devi accessed by address. One transfer unit of data is transferred in one bus cycle.
- Channel functions: The transfer mode that can be specified depends on the channe
  - Channel 0: External request can be accepted.
  - Channel 1: External request can be accepted.
  - Channel 2: This channel has a source address reload function, which reloads a address every four transfers.

communications interface (IrDA and SCIF), A/D converter (A/D) and a timer (C
request can be accepted in all the channels.)

— Auto request (The transfer request is generated automatically within the DMAC

- Selectable bus modes: Cycle-steal mode or burst mode
- Selectable channel priority levels:

  Fixed mode: The channel priority is fixed.

  Round-robin mode: The priority of the channel in which the execution request was
  made the lowest.
- Interrupt request: An interrupt request to the CPU can be generated after the specifi
  of transfers.

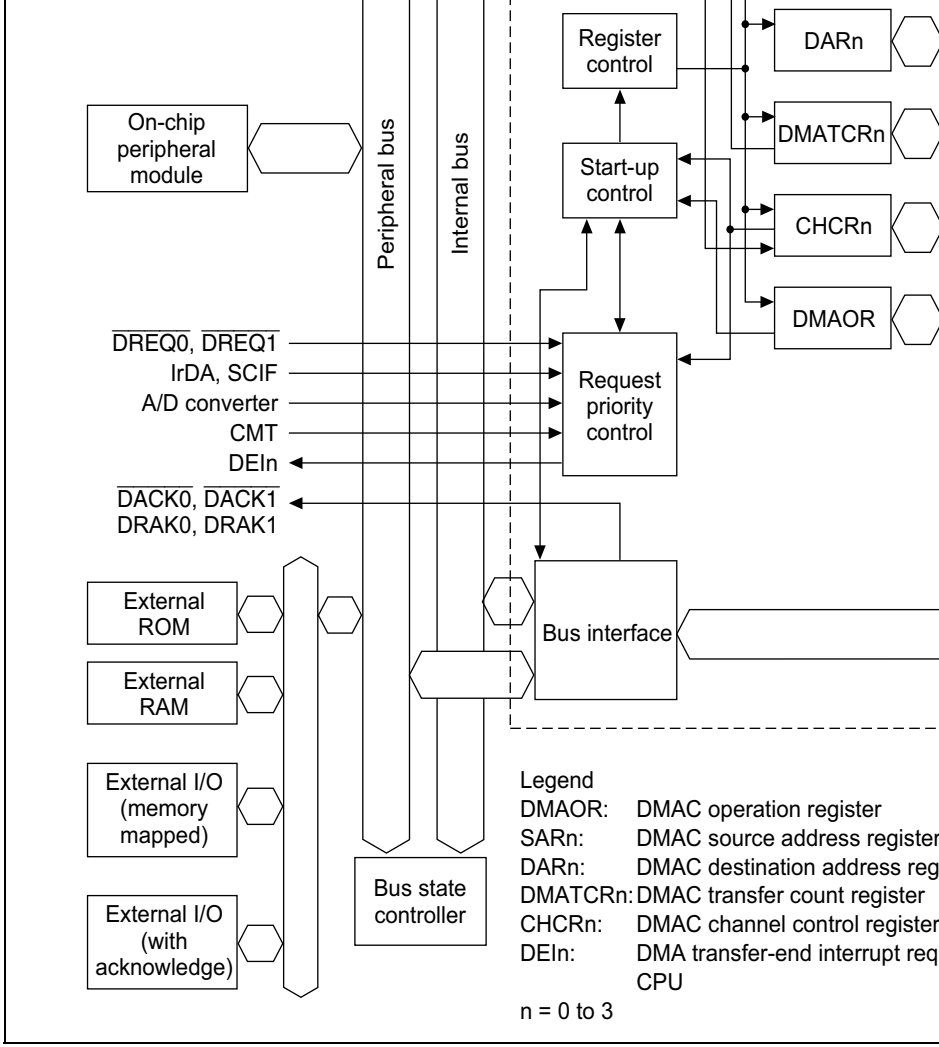


Figure 11.1 Block Diagram of DMAC

	DMA transfer request acceptance	DACK0	O	Strobe output to an externa DMA transfer request from device to channel 0
	DMA request acknowledge	DRAK0	O	Output showing that DREQ accepted
1	DMA transfer request	$\overline{\text{DREQ1}}$	I	DMA transfer request input external device to channel 1
	DMA transfer request acceptance	DACK1	O	Strobe output to an externa DMA transfer request from device to channel 1
	DMA request acknowledge	DRAK1	O	Output showing that DREQ accepted

	register 0				(H'A4000020)*4	
	DMA destination address register 0	DAR0	R/W	Undefined	H'04000024 (H'A4000024)*4	32
	DMA transfer count register 0	DMATCR0	R/W	Undefined	H'04000028 (H'A4000028)*4	24
	DMA channel control register 0	CHCR0	R/W*1	H'00000000	H'0400002C (H'A400002C)*4	32
1	DMA source address register 1	SAR1	R/W	Undefined	H'04000030 (H'A4000030)*4	32
	DMA destination address register 1	DAR1	R/W	Undefined	H'04000034 (H'A4000034)*4	32
	DMA transfer count register 1	DMATCR1	R/W	Undefined	H'04000038 (H'A4000038)*4	24
	DMA channel control register 1	CHCR1	R/W*1	H'00000000	H'0400003C (H'A400003C)*4	32
2	DMA source address register 2	SAR2	R/W	Undefined	H'04000040 (H'A4000040)*4	32
	DMA destination address register 2	DAR2	R/W	Undefined	H'04000044 (H'A4000044)*4	32
	DMA transfer count register 2	DMATCR2	R/W	Undefined	H'04000048 (H'A4000048)*4	24
	DMA channel control register 2	CHCR2	R/W*1	H'00000000	H'0400004C (H'A400004C)*4	32

	register 3				(H'A400005C)*4	
Shared	DMA operation register	DMAOR	R/W*1	H'0000	H'04000060 (H'A4000060)*4	16

Notes: These registers are located in area 1 of physical space. Therefore, when the ca either access these registers from the P2 area of logical space or else make an setting using the MMU so that these registers are not cached.

1. Only 0 can be written to bit 1 of CHCR0 to CHCR3, and bits 1 and 2 of DMA the flag after 1 is read.
2. If 16-bit access is used on SAR0 to SAR3, DAR0 to DAR3, and CHCR0 to C value in the 16 bits that were not accessed is retained.
3. DMATCR comprises the 24 bits from bit 0 to bit 23. The upper 8 bits, bits 24 cannot be written with 1 and are always read as 0.
4. When address translation by the MMU does not apply, the address in parent should be used.

To transfer data in 16 bits or in 32 bits, specify a 16-bit or 32-bit address boundary ad
transferring data in 16-byte units, a 16-byte boundary (address 16n) must be set for the
address value. Operation is not guaranteed if other addresses are specified.

An undefined value will be returned in a reset. The previous value is retained in standb

Bit:	31	30	29	28	27	26	25
Initial value:	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	23	22	21	20		…
						…
Initial value:	—	—	—	—		…
R/W:	R/W	R/W	R/W	R/W		…

An undefined value will be returned in a reset. The previous value is retained in standby

Bit:	31	30	29	28	27	26	25
Initial value:	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	23	22	21	20		…
						…
Initial value:	—	—	—	—		…
R/W:	R/W	R/W	R/W	R/W		…

Writing to upper eight bits in DMATCR is invalid; 0s are read if these bits are read. T
value should always be 0.

An undefined value will be returned in a reset. The previous value is retained in standb

Bit:	31	30	29	28	27	26	25
Initial value:	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R

Bit:	23	22	21	20	...
					...
Initial value:	—	—	—	—	...
R/W:	R/W	R/W	R/W	R/W	...

CHCR1; they are not used in CHCR2 and CHCR3. Consequently, writing to these bits in CHCR2 and CHCR3; 0s are read if these bits are read.

These register values are initialized to 0 in a reset. The previous value is retained in sta

Bit:	31	...	21	20	19	18	17
	—	...	—	DI	RO	RL	AM
Initial value:	0	...	0	0	0	0	0
R/W:	R	...	R	(R/W)*2	(R/W)*2	(R/W)*2	(R/W)*

Bit:	15	14	13	12	11	10	9
	DM1	DM0	SM1	SM0	RS3	RS2	RS1
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1
	—	DS	TM	TS1	TS0	IE	TE
Initial value:	0	0	0	0	0	0	0
R/W:	R	(R/W)*2	R/W	R/W	R/W	R/W	R/(W)*

Notes: 1. Only 0 can be written to the TE bit after 1 is read.
2. The DI, RO, RL, AM, AL, and DS bits are not included in some channels.

0	Direct address mode operation for channel 3	(
1	Indirect address mode operation for channel 3	

Bit 19—Source Address Reload Bit (RO): Selects whether the source address initial reloaded in channel 2.

This bit is only valid in CHCR2. Writing to this bit is invalid in CHCR0, CHCR1, an is read if this bit is read. The write value should always be 0. When using 16-byte tran must be cleared to 0, specifying non-reloading. Operation is not guaranteed if reloadin specified.

Bit 19: RO	Description	
0	Source address is not reloaded	(I
1	Source address is reloaded	

Bit 18—Request Check Level Bit (RL): Specifies whether DRAK ($\overline{\text{DREQ}}$ acknowle output is active-high or active-low.

This bit is only valid in CHCR0 and CHCR1. Writing to this bit is invalid in CHCR2 CHCR3; 0 is read if this bit is read. The write value should always be 0.

Bit 18: RL	Description	
0	Active-low DRAK output	(I
1	Active-high DRAK output	

0	DACK output in read cycle	(I
1	DACK output in write cycle	

Bit 16—Acknowledge Level (AL): Specifies whether DACK (acknowledge) signal ou active-high or active-low.

This bit is only valid in CHCR0 and CHCR1. Writing to this bit is invalid in CHCR2 a CHCR3; 0 is read if this bit is read. The write value should always be 0.

Bit 16: AL	Description	
0	Active-low DACK output	(In
1	Active-high DACK output	

Bits 15 and 14—Destination Address Mode Bits 1 and 0 (DM1, DM0): Select wheth DMA destination address is incremented, decremented, or left fixed.

Bit 15: DM1	Bit 14: DM0	Description	
0	0	Fixed destination address	(In
0	1	Destination address is incremented (+1 in 8-bit tran 16-bit transfer, +4 in 32-bit transfer, +16 in 16-byte t	
1	0	Destination address is decremented (–1 in 8-bit trar 16-bit transfer, –4 in 32-bit transfer; illegal setting in transfer)	
1	1	Setting prohibited	

		transfer)
1	1	Setting prohibited

If the transfer source is specified by indirect address, specify the address holding the v address in which the data to be transferred is stored (i.e. the indirect address) in source register 3 (SAR3).

Specification of SAR3 incrementing or decrementing in indirect address mode depend SM1 and SM0 settings. In this case, however, the SAR3 increment or decrement value or fixed at 0, regardless of the transfer data size specified in TS1 and TS0.

0	0	1	1	External request / Single address mode External device with DACK → external address
0	1	0	0	Auto request
0	1	0	1	Setting prohibited
0	1	1	0	Setting prohibited
0	1	1	1	Setting prohibited
1	0	0	0	Setting prohibited
1	0	0	1	Setting prohibited
1	0	1	0	IrDA transmission
1	0	1	1	IrDA reception
1	1	0	0	SCIF transmission
1	1	0	1	SCIF reception
1	1	1	0	A/D converter
1	1	1	1	CMT

Notes: When using 16-byte transfer, the following settings must not be made:

1010 IrDA transmission
1011 IrDA reception
1100 SCIF transmission
1101 SCIF reception
1110 A/D converter
1111 CMT

Operation is not guaranteed if these settings are made.

* External request specification is valid only in channels 0 and 1. None of the re sources can be selected in channels 2 and 3.

Bit 6: DS	Description
0	$\overline{\text{DREQ}}$ detected by low level
1	$\overline{\text{DREQ}}$ detected at falling edge

Bit 5—Transmit Mode (TM): Specifies the bus mode when transferring data.

Bit 5: TM	Description
0	Cycle-steal mode
1	Burst mode

Bits 4 and 3—Transmit Size Bits 1 and 0 (TS1, TS0): Specify the size of data to be

Bit 4: TS1	Bit 3: TS0	Description
0	0	Byte size (8 bits)
0	1	Word size (16 bits)
1	0	Longword size (32 bits)
1	1	16-byte unit (4 longword transfers)

Bit 2—Interrupt Enable Bit (IE): If this bit is set to 1, an interrupt is requested on c
the number of data transfers specified in DMATCR (i.e. when TE = 1).

Bit 2: IE	Description
0	Interrupt request is not generated on completion of data tra specified in DMATCR
1	Interrupt request is generated on completion of data transfe in DMATCR

	Clearing conditions: Writing 0 to TE after reading TE = 1 Power-on reset, manual reset
1	Data transfers specified in DMATCR completed

Bit 0—DMAC Enable Bit (DE): Enables operation of the corresponding channel.

Bit 0: DE	Description
0	Channel operation disabled (In
1	Channel operation enabled

If an auto-request is specified (RS3 to RS0), transfer starts when this bit is set to 1. In a request or an internal module request, transfer starts when a transfer request is generate bit is set to 1. Clearing this bit during transfer terminates the transfer.

Even if the DE bit is set, transfer is not enabled if the TE bit is 1, the DME bit in DMA the NMIF or AE bit in DMAOR is 1.

Bit:	7	6	5	4	3	2	1
	—	—	—	—	—	AE	NMI
Initial value:	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/(W)*	R/(W

Note: * Only 0 can be written to the AE and NMIF bits after 1 is read.

Bits 15 to 10—Reserved: These bits are always read as 0. The write value should alw

Bits 9 and 8—Priority Mode Bits 1 and 0 (PR1, PR0): Select the priority level betw channels when there are simultaneous transfer requests for multiple channels.

Bit 9: PR1	Bit 8: PR0	Description
0	0	CH0 > CH1 > CH2 > CH3
0	1	CH0 > CH2 > CH3 > CH1
1	0	CH2 > CH0 > CH1 > CH3
1	1	Round-robin

Bits 7 to 3—Reserved: These bits are always read as 0. The write value should alway

1	DMAC address error; DMA transfer is disabled
	This bit is set by occurrence of a DMAC address error

Bit 1—NMI Flag Bit (NMIF): Indicates that an NMI is input. This bit is set regardless whether the DMAC is in the operating or halted state. The CPU cannot write 1 to this b can be written to clear this bit after 1 is read.

Bit 1: NMIF	Description
0	No NMI input; DMA transfer is enabled (In
	Clearing conditions: Writing 0 to NMIF after reading NMIF =
	Power-on reset, manual reset
1	NMI input; DMA transfer is disabled
	This bit is set by occurrence of an NMI interrupt

Bit 0—DMA Master Enable Bit (DME): Enables or disables the DMAC on all chann DME bit and the DE bit corresponding to each channel in CHCR are set to 1, transfer is on the corresponding channel. If this bit is cleared during transfer, transfer on all the ch be terminated.

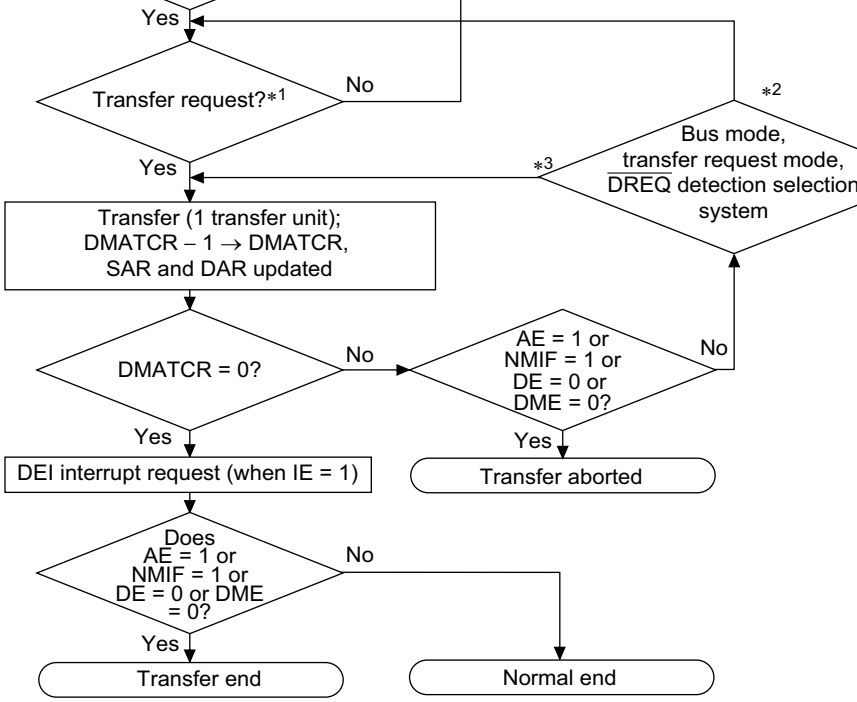
Even if the DME bit is set, transfer is not enabled if the TE bit is 1 or the DE bit is 0 in the NMIF or AE bit is 1 in DMAOR.

Bit 0: DME	Description
0	DMA transfer disabled on all channels (In
1	DMA transfer enabled on all channels

After the DMA source address register (SAR), DMA destination address register (DA
transfer count register (DMATCR), DMA channel control register (CHCR), and DMA
register (DMAOR) are set, the DMAC transfers data according to the following proce

1. Checks to see if transfer is enabled (DE = 1, DME = 1, TE = 0, AE = 0, NMIF = 0
2. When a transfer request comes and transfer is enabled, the DMAC transfers 1 trans
   data (according to the TS0 and TS1 settings). For an auto-request, the transfer beg
   automatically when the DE bit and DME bit are set to 1. The DMATCR value wil
   decremented for each transfer. The actual transfer flows vary by address mode and
3. When the specified number of transfers have been completed (when DMATCR rea
   transfer ends normally. If the IE bit in CHCR is set to 1 at this time, a DEI interrup
   the CPU.
4. When an address error occurs by the DMAC or an NMI interrupt is generated, the
   aborted.

Figure 11.2 is a flowchart of this procedure.



Notes: 1. In auto-request mode, transfer begins when AE, NMIF, and TE are both 0 and the I
DME bits are set to 1.
2. $\overline{\text{DREQ}}$ = level detection in burst mode (external request) or cycle-steal mode.
3. $\overline{\text{DREQ}}$ = edge detection in burst mode (external request), or auto-request mode in

Figure 11.2 DMAC Transfer Flowchart

memory-to-memory transfer or a transfer between memory and an on-chip peripheral unable to request a transfer, the auto-request mode allows the DMAC to automatically transfer request signal internally. When the DE bit of CHCR0–CHCR3 and the DME DMAOR are set to 1, the transfer begins so long as the TE bit of CHCR0–CHCR3 and and AE bits of DMAOR are 0.

External Request Mode: In this mode a transfer is performed in response to the requ ($\overline{\text{DREQ}}$) of an external device. Choose one of the modes shown in table 11.3 accordin application system. When this mode is selected, if DMA transfer is enabled (DE = 1, TE = 0, AE = 0, NMIF = 0), a transfer is performed upon a request at the $\overline{\text{DREQ}}$ input $\overline{\text{DREQ}}$ detection by either a falling edge or low level of the signal input with the DS b and CHCR1 (DS = 0 for level detection, DS = 1 for edge detection). The source of the request does not have to be the data transfer source or destination.

Table 11.3 Selecting External Request Modes with RS Bits

RS3	RS2	RS1	RS0	Address Mode	Source	Destin
0	0	0	0	Dual address mode	Any*	Any*
		1	0	Single address mode	External memory, memory-mapped external device	Extern with D
			1		External device with DACK	Extern memo extern

Note: * External memory, memory-mapped external device, on-chip memory, on-chip module (This applies only to IrDA, SCIF, A/D converter, D/A converter, and I/O

On-Chip Module Request Mode: In this mode a transfer is performed in response to request signal (interrupt request signal) of an on-chip module. This mode cannot be se 16-byte transfer. These are six transfer request signals: the receive-data-full interrupts the transmit-data-empty interrupts (TXI) from two serial communication interfaces (I the A/D conversion end interrupt (ADI) of the A/D converter, and the compare match interrupt (CMI) of the CMT (table 11.4). When this mode is selected, if DMA transfer (DE = 1, DME = 1, TE = 0, AE = 0, NMIF = 0), a transfer is performed upon input of

RS3	RS2	RS1	RS0	Request Source	DMA Transfer Request Signal	Source	Destination
1	0	1	0	IrDA transmitter	TXI1 (IrDA transmit-data-empty interrupt transfer request)	Any*	TDR1
1	0	1	1	IrDA receiver	RXI1 (IrDA receive-data-full interrupt transfer request)	RDR1	Any*
1	1	0	0	SCIF transmitter	TXI2 (SCIF transmit-data-empty interrupt transfer request)	Any*	TDR2
1	1	0	1	SCIF receiver	RXI2 (SCIF receive-data-full interrupt transfer request)	RDR1	Any*
1	1	1	0	A/D converter	ADI (A/D conversion end interrupt)	ADDR	Any*
1	1	1	1	CMT	CMI (Compare match timer interrupt)	Any*	Any*

ADDR: A/D data register of A/D converter

Note: * External memory, memory-mapped external device, on-chip peripheral module applies only to IrDA, SCIF, A/D converter, D/A converter, and I/O ports.)

When outputting transfer requests from on-chip peripheral modules, the appropriate int enable bits must be set to output the interrupt signals.

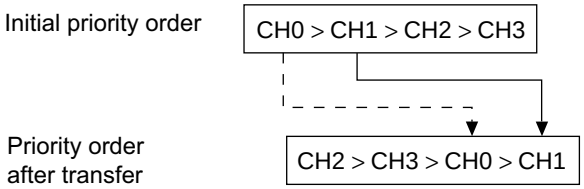
If the interrupt request signal of the on-chip peripheral module is used as a DMA transf signal, an interrupt is not sent to the CPU.

The DMA transfer request signals in table 11.4 are automatically discontinued when th corresponding DMA transfer is performed. If cycle-steal mode is being employed, they withdrawn at the first transfer; if burst mode is being used, they are discontinued at the transfer.

CH0 > CH2 > CH3 > CH1
CH2 > CH0 > CH1 > CH3

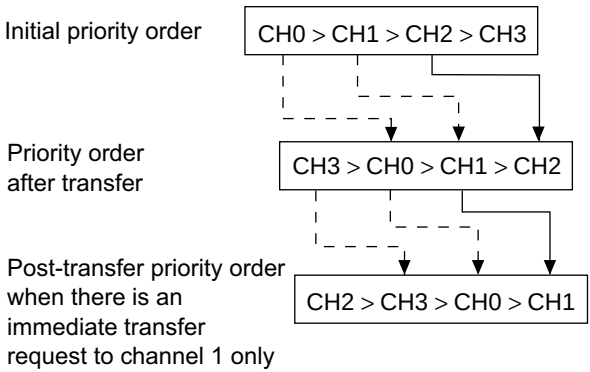
These are selected by the PR1 and PR0 bits in DMAOR.

Round-Robin Mode: Each time one word, byte, or longword is transferred on one ch
priority order is rotated. The channel on which the transfer was just finished rotates to
of the priority order. The round-robin mode operation is shown in figure 11.3. The pr
round-robin mode is CH0 > CH1 > CH2 > CH3 immediately after reset.



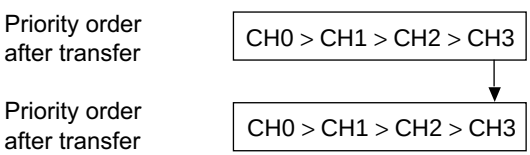
Channel 1 becomes lowes
priority.
The priority of channel 0, v
was higher than channel 1
shifted.

(3) When channel 2 transfers



Channel 2 becomes lowes
priority.
The priority of channels 0 a
which were higher than ch
are also shifted. If immedia
after there is a request to t
channel 1 only, channel 1
lowest-priority and the prio
channels 3 and 0, which w
higher than channel 1, is a
shifted.

(4) When channel 3 transfers



Priority order does not cha

Figure 11.3 Round-Robin Mode

4. When the channel 0 transfer ends, channel 0 becomes lowest-priority.
5. At this point, channel 1 has a higher priority than channel 3, so the channel 1 trans (channel 3 waits for transfer).
6. When the channel 1 transfer ends, channel 1 becomes lowest-priority.
7. The channel 3 transfer begins.
8. When the channel 3 transfer ends, channels 3 and 2 shift downward in priority so t 3 becomes the lowest-priority.

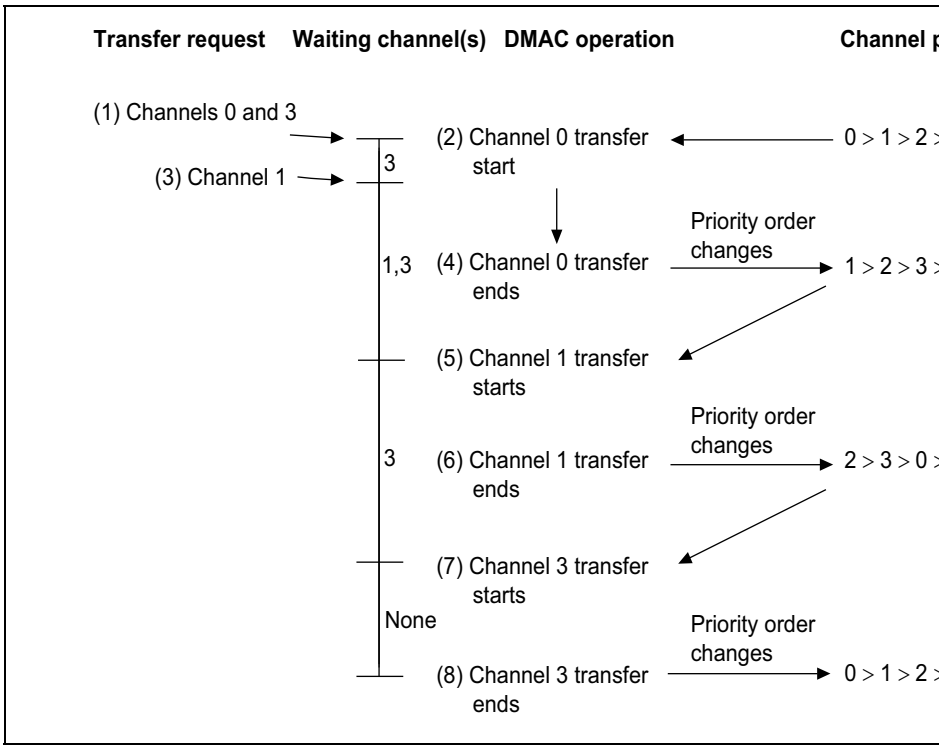


Figure 11.4 Changes in Channel Priority in Round-Robin Mode

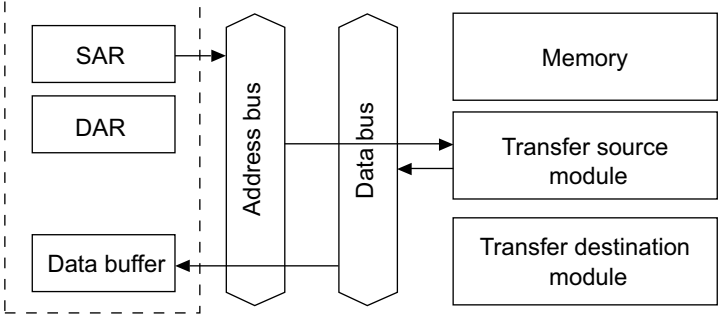
	Destination			
Source	**External Device with DACK**	**External Memory**	**Memory-Mapped External Device**	**On-C Perip Modu**
External device with DACK	Not available	Dual, single	Dual, single	Not a
External memory	Dual, single	Dual	Dual	Dual
Memory-mapped external device	Dual, single	Dual	Dual	Dual
On-chip peripheral module	Not available	Dual	Dual	Dual

Notes: 1. Dual: Dual address mode
2. Single: Single address mode
3. Dual address mode includes direct address mode and indirect address mode
4. 16-byte transfer is not available for on-chip peripheral modules.

Address Modes:

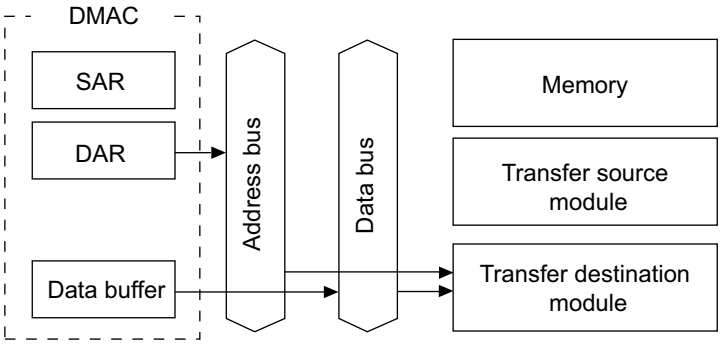
- Dual Address Mode

  In dual address mode, both the transfer source and destination are accessed (selectal address. The source and destination can be located externally or internally. Dual add has (1) a direct address transfer mode and (2) an indirect address transfer mode.



The SAR value is an address, data is read from the transfer source module and the data is temporarily stored in the DMAC.

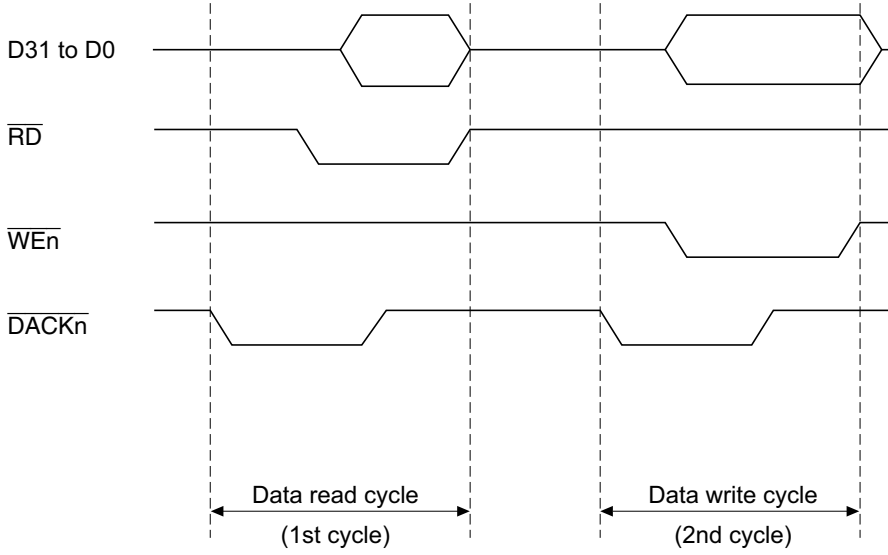
First bus cycle



The DAR value is an address, and the value stored in the data buffer in the DMAC is written to the transfer destination module.

Second bus cycle

Figure 11.5 Operation of Direct Address Mode in Dual Address Mod



Note: In transfer between external memories, with DACK output in the read cycle, I
output timing is the same as that of CSn.

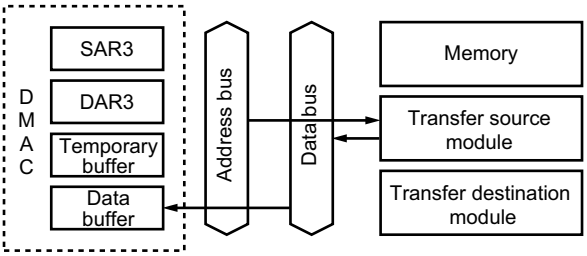
Figure 11.6 Example of DMA Transfer Timing in the Direct Address Mode in D
(Transfer Source: Ordinary Memory, Transfer Destination: Ordinary Mem

(2) In indirect address transfer mode, the address of memory in which data to be tra
stored is specified in the transfer source address register (SAR3) in the DMAC.
Consequently, in this mode, the address value specified in the transfer source ad
register in the DMAC is read first. This value is temporarily stored in the DMA
the read value is output as an address, and the value stored in that address is sto
DMAC again. Then, the value read afterwards is written to the address specifie
transfer destination address; this completes one DMA transfer. 16-byte transfer
possible.

Figure 11.7 shows an example. In this example, the transfer destination, the tra
source, and the storage destination of the indirect address are 16-bit external me
and transfer data is 16 or 8 bits. Figure 11.8 shows an example of the transfer t

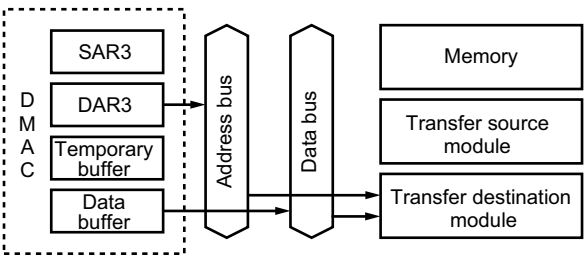
the value is stored in the temporary buffer. The value to be read must be 32 bits since it is used for the address. If data bus connected to an external memory space is 16 bits wide, two bus cycles are necessary.

First and second bus cycles



When the value in the temporary buffer is an address, the data is read from the transfer source module to the data buffer.

Third bus cycle

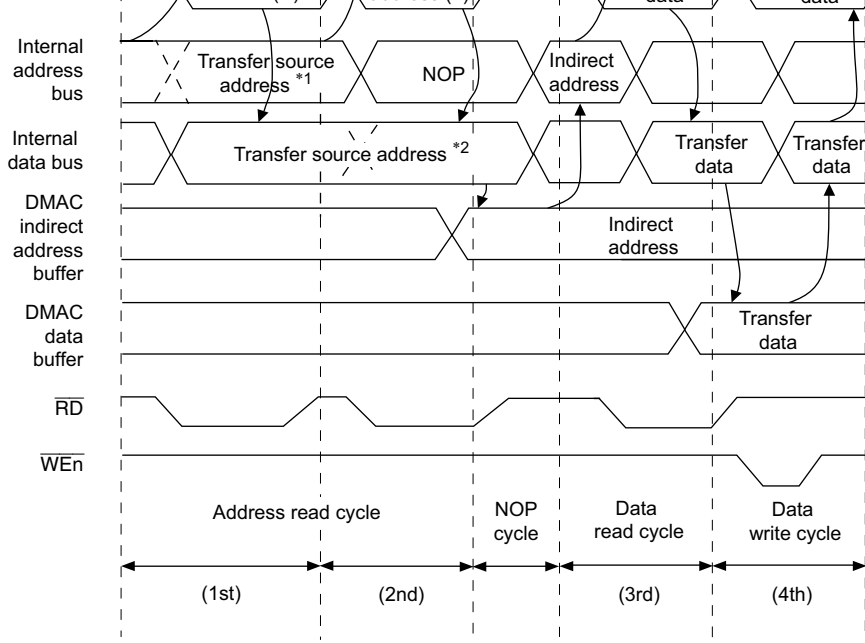


When the value in SAR3 is an address, the value in the data buffer is written to the transfer source module.

Fourth bus cycle

Note: This example shows memory, the transfer source module, and the transfer destination module; in practice, any module can be connected in the addressing space.

Figure 11.7 Indirect Address Operation in Dual Address Mode (When External Memory Space has a 16-Bit Width)



External memory space → external memory space (external memory is 16-bit width)

Notes: 1. The internal address bus value does not change, and is controlled by the port.
2. The DMAC does not fetch the value until 32-bit data is output to the internal data bu

Figure 11.8 Example of Transfer Timing in the Indirect Address Mode in Dual Address Mode

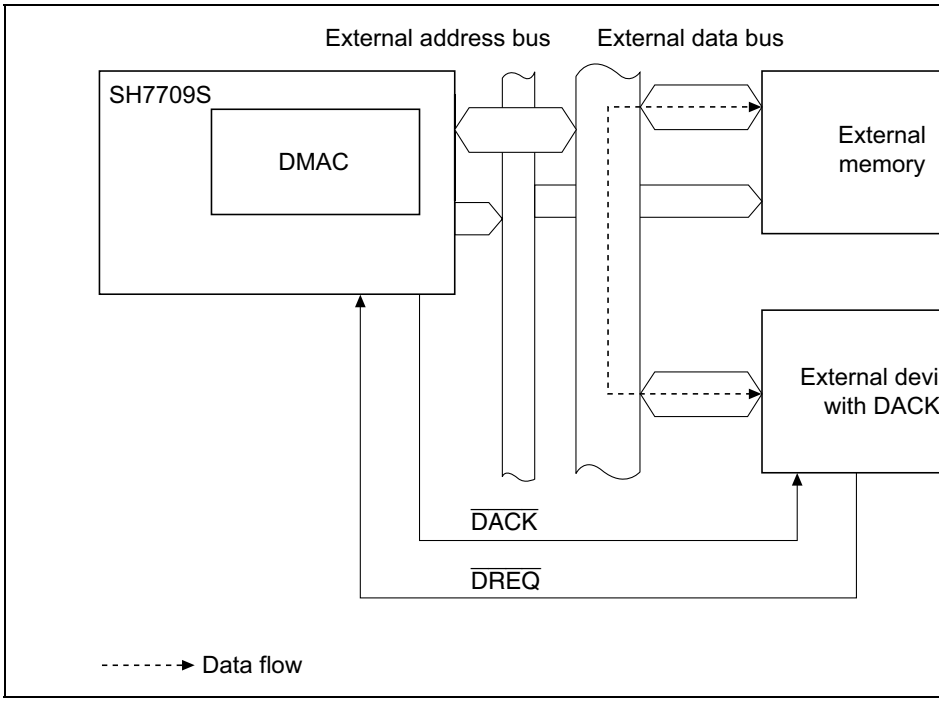


Figure 11.9 Data Flow in Single Address Mode

Two kinds of transfer are possible in single address mode: (1) transfer between an
device with DACK and a memory-mapped external device, and (2) transfer betwe
external device with DACK and external memory. In both cases, only the external
signal ($\overline{\text{DREQ}}$) is used for transfer requests.

Figures 11.10 and 11.11 show examples of DMA transfer timing in single address

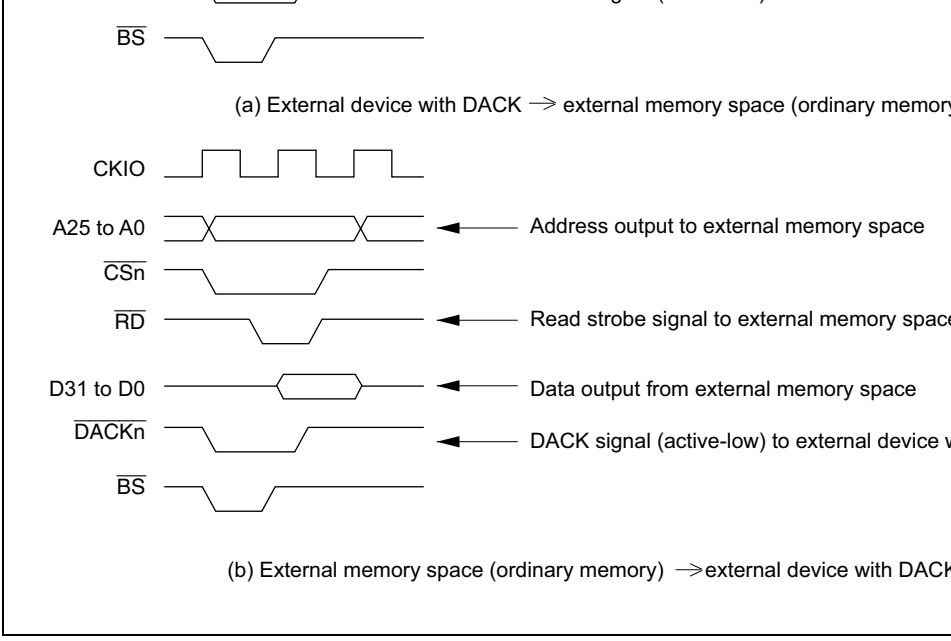


Figure 11.10 Example of DMA Transfer Timing in Single Address Mod

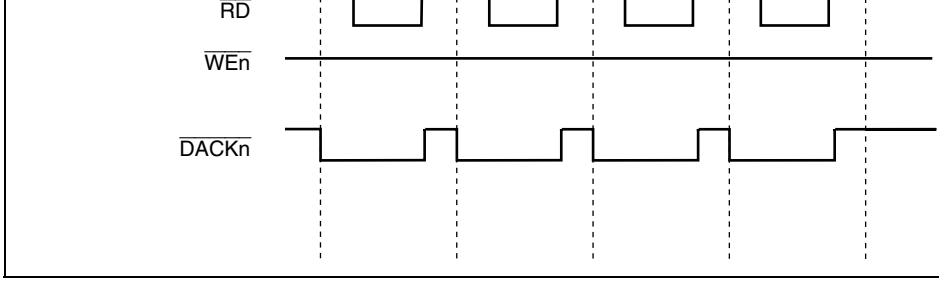


Figure 11.11 Example of DMA Transfer Timing in Single Address Mo
(16-byte Transfer, External Memory Space (Ordinary Memory) → External D
DACK)

Bus Modes: There are two bus modes: cycle-steal and burst. Select the mode in the T
CHCR0–CHCR3.

- Cycle-Steal Mode

  In cycle-steal mode, the bus is given to another bus master after a one-transfer-uni
  word, longword, or 16-byte unit) DMAC. When another transfer request occurs, th
  obtained from the other bus master and transfer is performed for one transfer unit.
  transfer ends, the bus is passed to the other bus master. This is repeated until the tr
  conditions are satisfied.

  In the cycle-steal mode, transfer areas are not affected regardless of the transfer re
  transfer source, and transfer destination settings. Figure 11.12 shows an example o
  transfer timing in cycle-steal mode. Transfer conditions shown in the figure are:

  Dual address mode
  $\overline{\text{DREQ}}$ level detection

- Burst Mode

  Once the bus is obtained, the transfer is performed continuously until the transfer en condition is satisfied. In external request mode with low level detection of the $\overline{\text{DRE}}$ however, when the $\overline{\text{DREQ}}$ pin is driven high, the bus passes to the other bus master DMAC transfer request that has already been accepted ends, even if the transfer end have not been satisfied.

  Burst mode cannot be used when a serial communication interface (IrDA, SCI), or A converter is the transfer request source. Figure 11.13 shows an example of burst mo

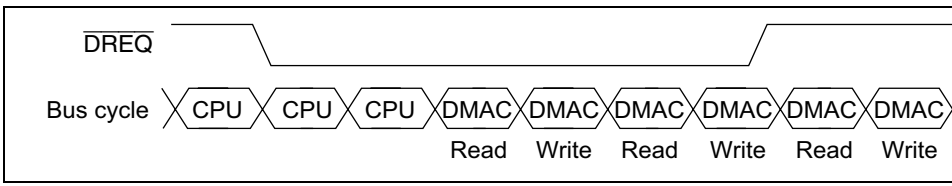


Figure 11.13 Example of Transfer in Burst Mode

	External device with DACK and memory-mapped external device	External	B/C	8/16/32/128
	External memory and external memory	All[*1]	B/C	8/16/32/128
	External memory and memory-mapped external device	All [*1]	B/C	8/16/32/128
	Memory-mapped external device and memory-mapped external device	All [*1]	B/C	8/16/32/128
	External memory and on-chip peripheral module	All [*2]	B/C[*3]	8/16/32[*4]
	Memory-mapped external device and on-chip peripheral module	All [*2]	B/C[*3]	8/16/32[*4]
	On-chip peripheral module and on-chip peripheral module	All [*2]	B/C[*3]	8/16/32[*4]
Single	External device with DACK and external memory	External	B/C	8/16/32/128
	External device with DACK and memory-mapped external device	External	B/C	8/16/32/128

B: Burst, C: Cycle-steal

Notes: 1. External requests, auto requests and on-chip peripheral module (CMT) req available.

2. External requests, auto requests and on-chip peripheral module requests a available. When the IrDA, SCIF, or A/D converter is also the transfer reque however, the transfer destination or transfer source must be the IrDA, SCIF converter, respectively.

3. If the transfer request source is the IrDA, SCIF, or A/D converter only cycle is available.

4. The access size permitted when the transfer destination or source is an on peripheral module register.

5. If the transfer request is an external request, only channels 0 and 1 are ava

completes the transfer of one transfer unit, even if channel 0 is in cycle-steal mode or b
The bus will then switch between the two in the order channel 1, channel 0, channel 1,

Even if the priority is set in fixed mode or in round-robin mode, the bus will not be give
CPU since channel 1 is in burst mode. This example is illustrated in figure 11.14.

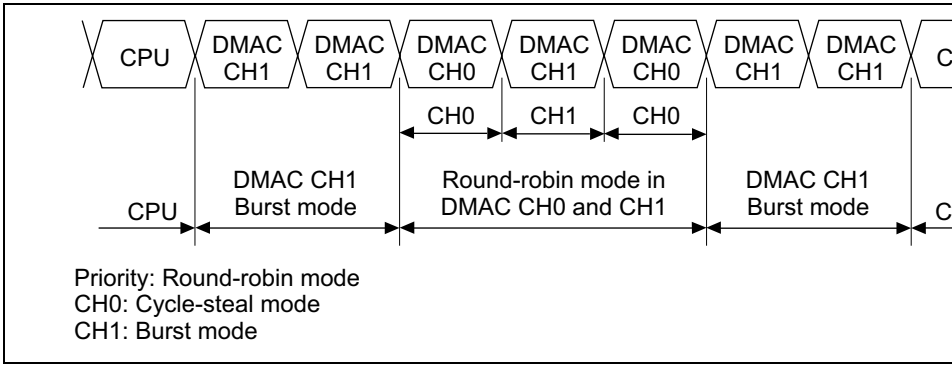


Figure 11.14 Bus State when Multiple Channels Are Operating

The second and subsequent $\overline{\text{DREQ}}$ sampling operations are started two cycles after the sample.

Operation

- Cycle-Steal Mode

  In cycle-steal mode, the $\overline{\text{DREQ}}$ sampling timing is the same regardless of whether edge detection is used.

  For example, in figure 11.15 (cycle-steal mode, level input), DMAC transfer begin earliest, three cycles after the first sampling is performed. The second sampling is cycles after the first. If $\overline{\text{DREQ}}$ is not detected at this time, sampling is performed in subsequent cycle.

  Thus, $\overline{\text{DREQ}}$ sampling is performed one step in advance. The third sampling opera performed until the idle cycle following the end of the first DMA transfer.

  The above conditions are the same whatever the number of CPU transfer cycles, as figure 11.16. The above conditions are also the same whatever the number of DMA cycles, as shown in figure 11.17.

  DACK is output in a read in the example in figure 11.15, and in a write in the exam figure 11.16. In both cases, DACK is output for the same duration as $\overline{\text{CS}}$n.

  Figure 11.18 shows an example in which sampling is executed in all subsequent cy $\overline{\text{DREQ}}$ cannot be detected.

In burst mode, also, the DACK output period is the same as in cycle-steal mode.

- Burst Mode, Edge Detection
  In the case of burst mode with edge detection, $\overline{\text{DREQ}}$ sampling is only performed o

  For example, in figure 11.21, DMAC transfer begins, at the earliest, three cycles aft
  sampling is performed. After this, DMAC transfer is executed continuously until th
  of data transfers set in the DMATCR register have been completed. $\overline{\text{DREQ}}$ is not sa
  during this time.

  To restart DMAC after it has been suspended by an NMI, first clear NMIF, then inp
  request again.

  In burst mode, also, the DACK output period is the same as in cycle-steal mode.

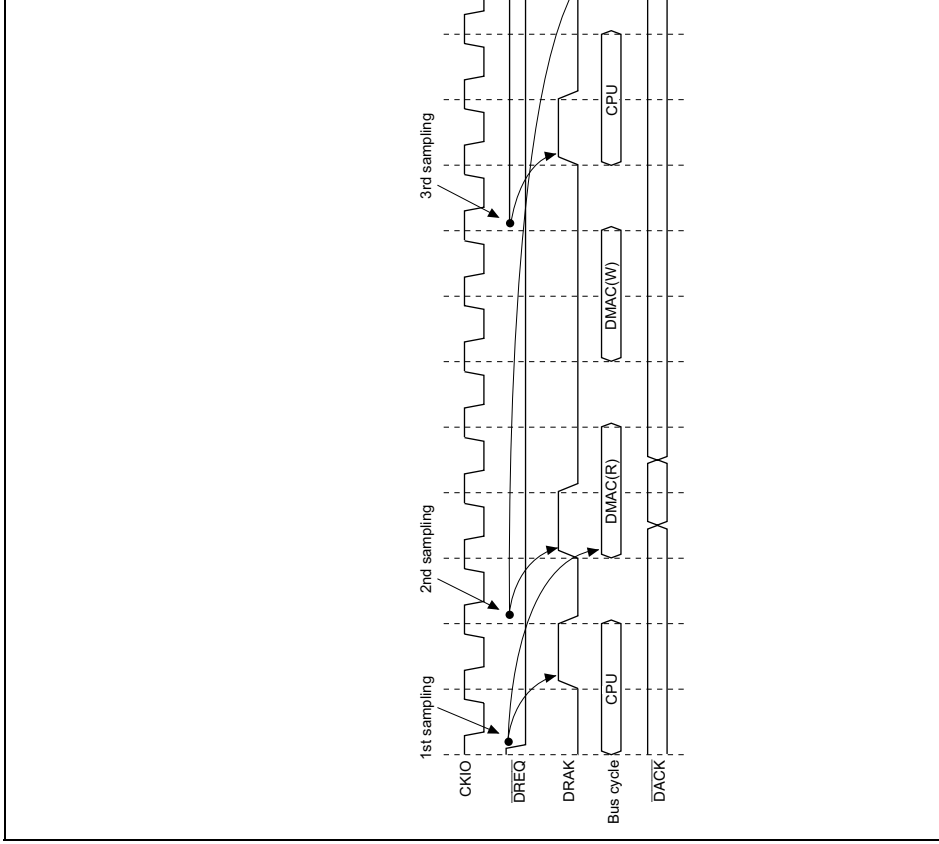


Figure 11.15 Cycle-Steal Mode, Level Input (CPU Access: 2 Cycles)

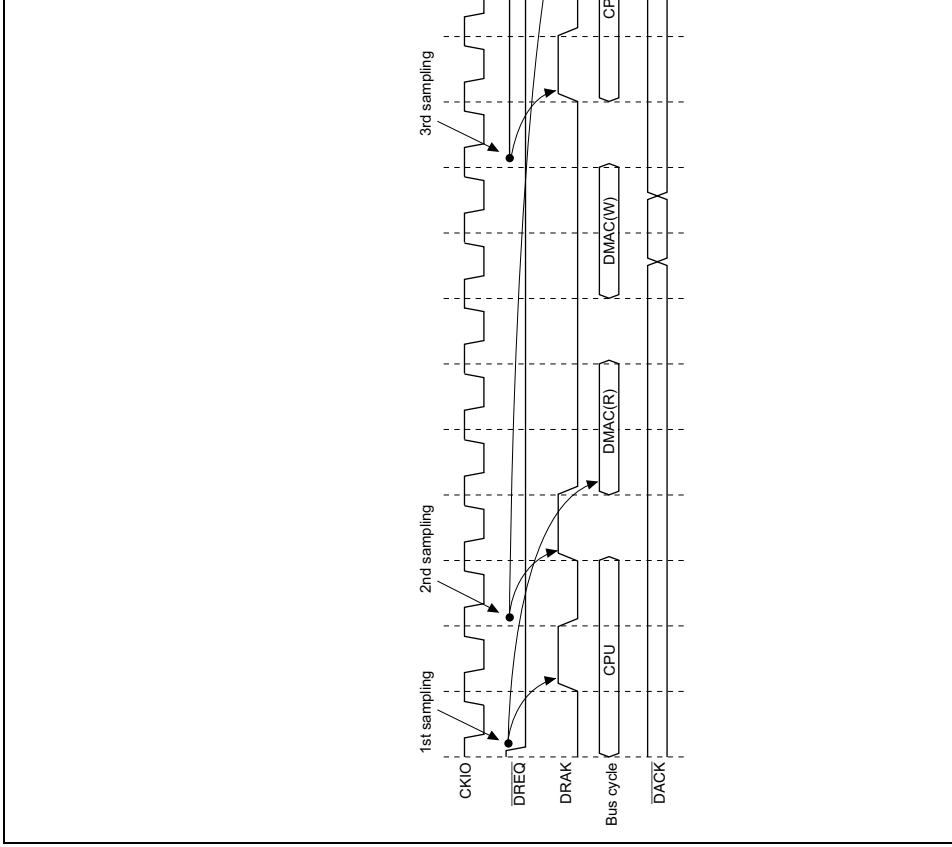


Figure 11.16 Cycle-Steal Mode, Level Input (CPU Access: 3 Cycles)

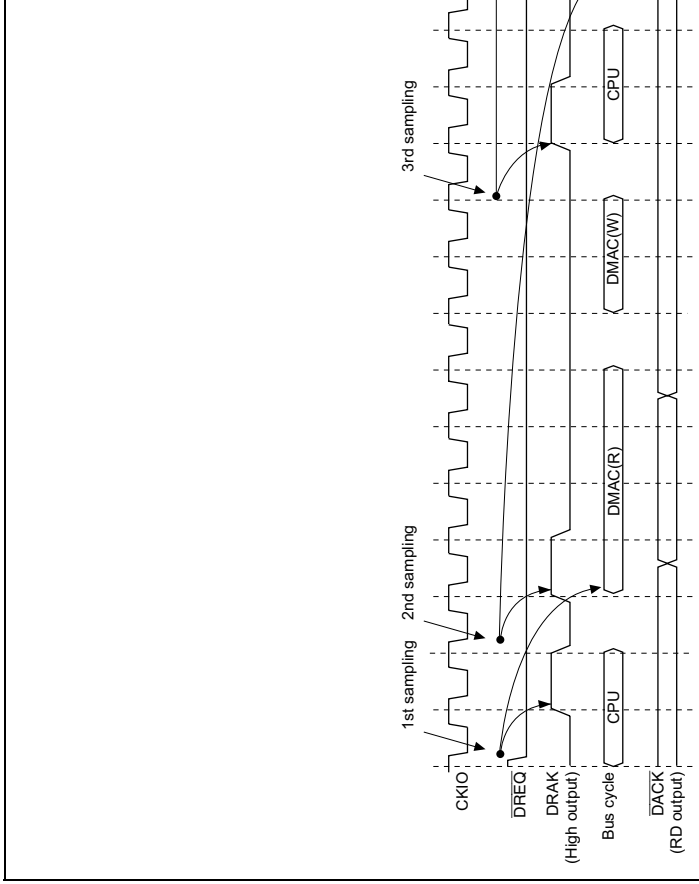


Figure 11.17 Cycle-Steal Mode, Level input (CPU Access: 2 Cycles, DMA RI
4 Cycles)

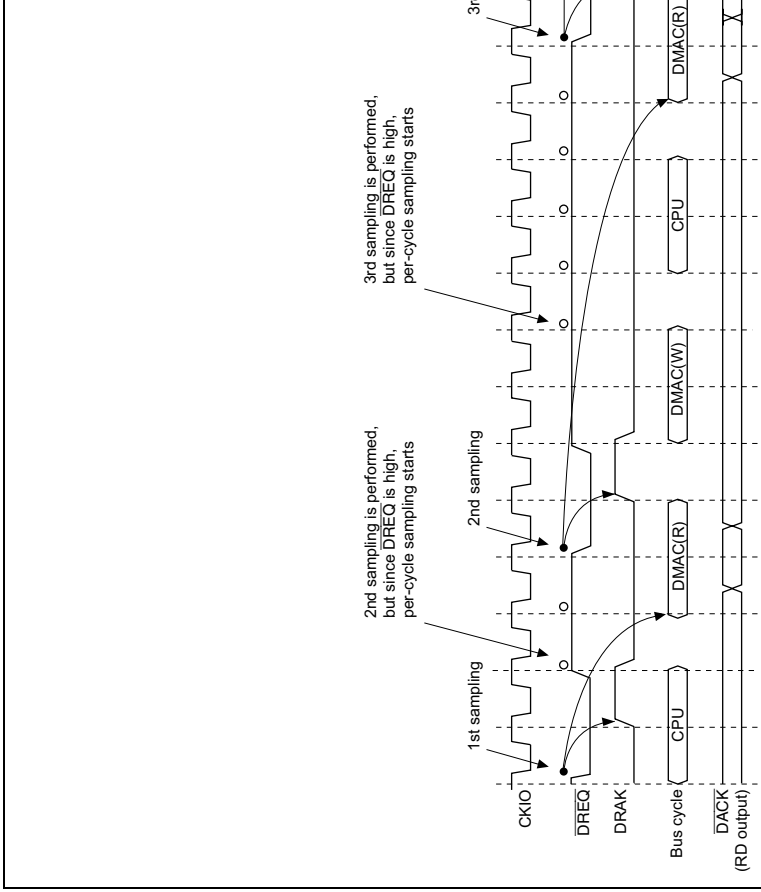


Figure 11.18 Cycle-Steal Mode, Level input (CPU Access: 2 Cycles, $\overline{\text{DREQ}}$ Inpu

Figure 11.19 Cycle-Steal Mode, Edge input (CPU Access: 2 Cycles)

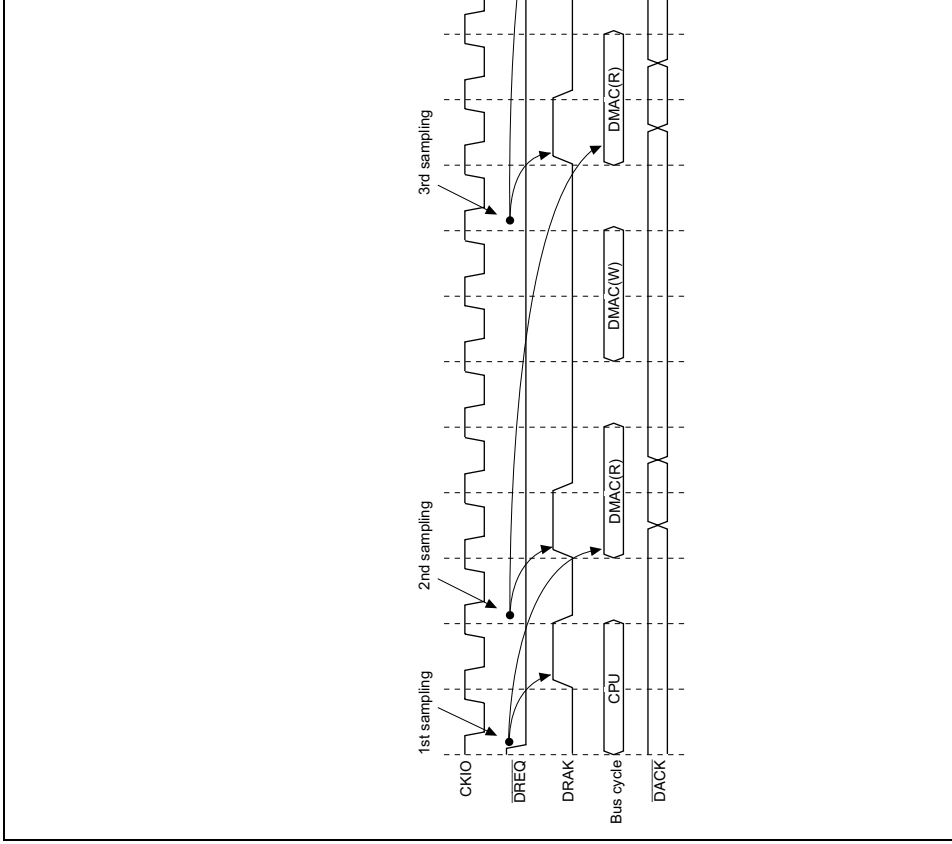


Figure 11.20 Burst Mode, Level Input

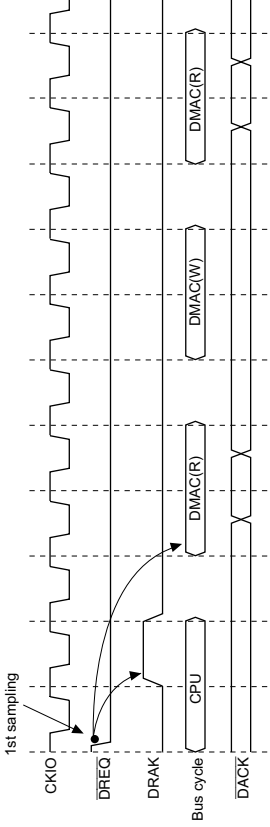


Figure 11.21 Burst Mode, Edge Input

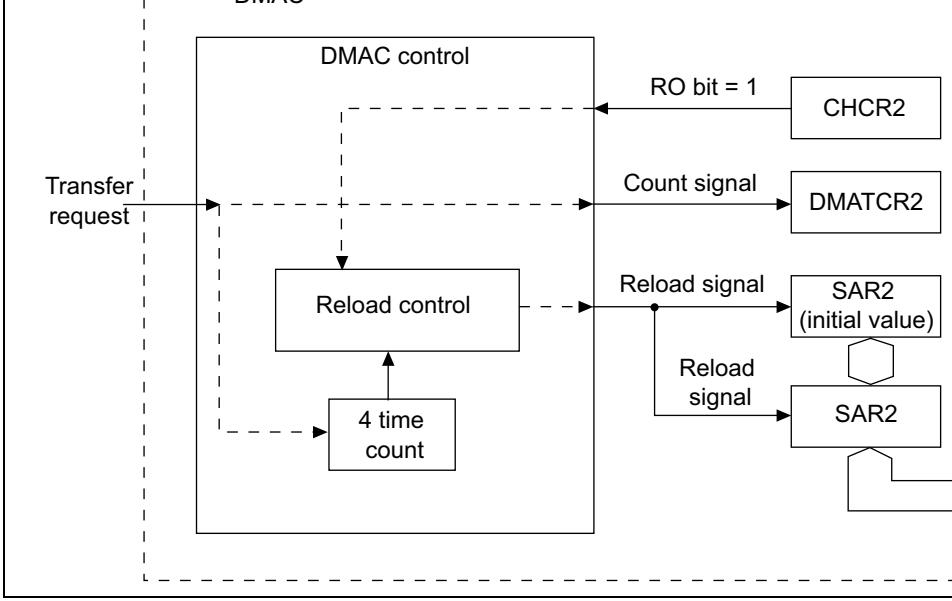


Figure 11.22 Source Address Reload Function Diagram

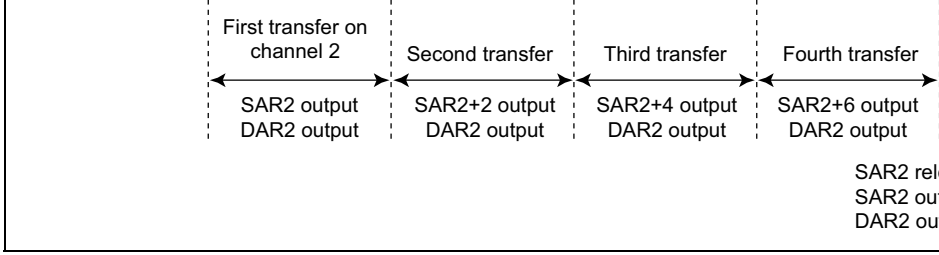


Figure 11.23 Timing Chart of Source Address Reload Function

The reload function can be executed with a transfer data size of 8, 16, or 32 bits.

DMATCR2, which specifies the transfer count, decrements 1 each time a transfer ends of whether the reload function is on or off. Consequently, a multiple of four must be s DMATCR2 when the reload function is on. Operation is not guaranteed if other values specified.

The counter that counts the execution of four transfers for the address reload function clearing the DME bit in DMAOR or the DE bit in CHCR2, by setting the transfer end in CHCR2), by DMAC address error, and by NMI input, as well as by a reset, but the DAR2, and DMATCR2 registers are not reset. Therefore, if these sources are generate be a mix of an initialized counter and uninitialized registers in the DMAC, and a malf be caused by restarting the DMAC in that state. Consequently, if one of these sources setting of the TE bit occurs during use of the address reload function, set SAR2, DAR DMATCR2 again.

accepted until the ending conditions are satisfied.

In cycle-steal mode, the operation is the same regardless of whether the transfer req
detected by level or edge.

(b) Burst mode, edge detection (external request, internal request, and auto-request)

The timing from the point where the ending conditions are satisfied to the point whe
DMAC stops operating is the same as in cycle-steal mode. With edge detection in b
though only one transfer request is generated to start the DMAC, stop request samp
performed at the same timing as transfer request sampling in cycle-steal mode. As a
period when a stop request is not sampled is regarded as the period when a transfer
generated, and after performing the DMA transfer for this period, the DMAC stops

(c) Burst mode, level detection (external request)

Same as in (a).

(d) Bus timing when transfer is suspended

Transfer is suspended when one transfer ends. Even if transfer ending conditions ar
during a read in direct address transfer in dual address mode, the subsequent write p
executed, and after the transfer in (a) to (c) above has been executed, DMAC operat
suspended.

Individual Channel Ending Conditions: There are two ending conditions. A transfer
the value of the channel's DMA transfer count register (DMATCR) is 0, or when the D
channel's CHCR register is cleared to 0.

- When DMATCR is 0: When the DMATCR value becomes 0 and the corresponding
DMA transfer ends, the transfer end flag bit (TE) is set in CHCR. If the IE (interrup
bit has been set, a DMAC interrupt (DEI) request is sent to the CPU. This transfer e
not apply to (a) to (d) described above.
- When DE in CHCR is 0: Software can halt a DMA transfer by clearing the DE bit i
channel's CHCR register. The TE bit is not set when this happens. This transfer end
to (a) to (d) described above.

bit to 0. At this time, if there are channels that should not be restarted, clear the co
DE bit in CHCR.

- Transfer ending when DME is cleared to 0 in DMAOR: Clearing the DME bit to 0
forcibly aborts transfer on all channels. The TE bit is not set. All channels abort th
according to the conditions in (a) to (d) in section 11.3.7, DMA Transfer Ending C
in NMI interrupt generation. In this case, the values in SAR, DAR, and DMATCR
updated.

The CMT has the following features:

- Four types of counter input clock can be selected
  — One of four internal clocks (Pϕ/4, Pϕ/8, Pϕ/16, Pϕ/64) can be selected.
- Generates a DMA transfer request when compare match occurs.

Block Diagram

Figure 11.24 shows a block diagram of the CMT.

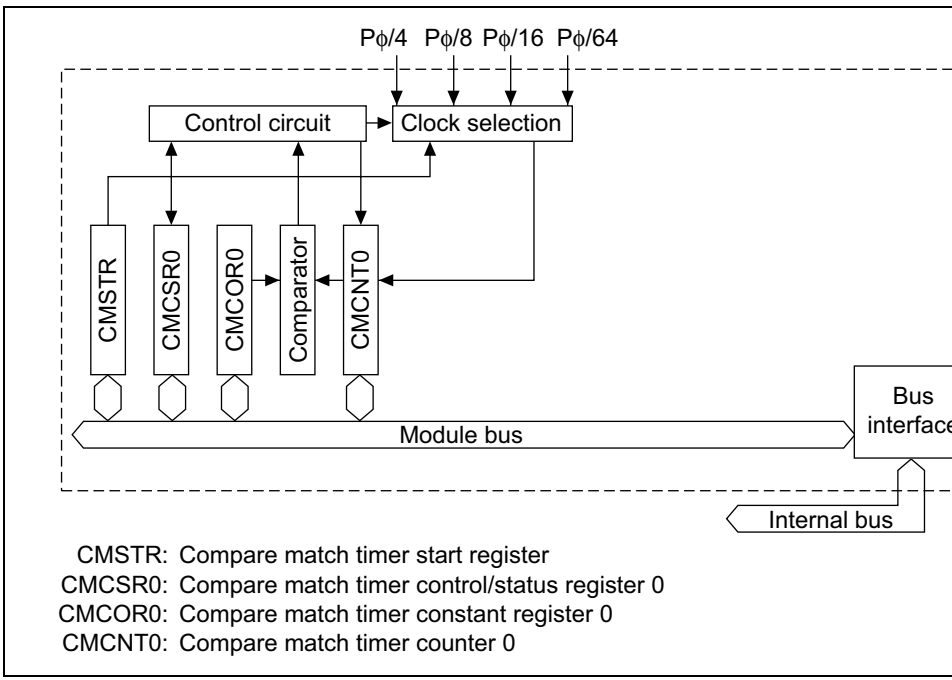


Figure 11.24 Block Diagram of CMT

Compare match timer control/status register 0	CMCSR0	R/(W)*1	H'0000	H'04000072 (H'A4000072)*2	8
Compare match counter 0	CMCNT0	R/W	H'0000	H'04000074 (H'A4000074)*2	8
Compare match constant register 0	CMCOR0	R/W	H'FFFF	H'04000076 (H'A4000076)*2	8

Notes: 1. The only value that can be written to the CMF bit in CMCSR0 is 0 to clear t
2. When address translation by the MMU does not apply, the address in parer should be used.

11.4.2 Register Descriptions

Compare Match Timer Start Register (CMSTR)

The compare match timer start register (CMSTR) is a 16-bit register that selects whetl match counter 0 (CMCNT0) is operated or halted. It is initialized to H'0000 by a reset its previous value in standby mode.

Bit:	15	14	13	12	11	10	9
	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R

Bit:	7	6	5	4	3	2	1
	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W

Bits 15 to 2—Reserved: These bits are always read as 0. The write value should alwa

Bit 1—Reserved: This bit can be read or written. The write value should always be 0.

occurrence of compare matches, sets the enable/disable status of interrupts, and establis
clock used for incrementation. It is initialized to H'0000 by a reset, but retains its previo
standby mode.

Bit:	15	14	13	12	11	10	9
	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R

Bit:	7	6	5	4	3	2	1
	CMF	—	—	—	—	—	CKS1
Initial value:	0	0	0	0	0	0	0
R/W:	R/(W)*	R/W	R	R	R	R	R/W

Note: * The only value that can be written is 0 to clear the flag.

Bits 15 to 8 and 5 to 2—Reserved: These bits are always read as 0. The write value sh
always be 0.

Bit 7—Compare Match Flag (CMF): Indicates whether or not the compare match tim
0 (CMCNT0) and compare match timer constant 0 (CMCOR0) values match.

Bit 7: CMF	Description
0	CMCNT0 and CMCOR0 values do not match (In
	Clearing condition: Write 0 to CMF after reading CMF = 1
1	CMCNT0 and CMCOR0 values match

1	0	P φ/16
	1	P φ/64

Compare Match Counter 0 (CMCNT0)

Compare match counter 0 (CMCNT0) is a 16-bit register used as an up-counter.

When an internal clock is selected with the CKS1 and CKS0 bits in the CMCSR0 regi
STR bit in CMSTR is set to 1, CMCNT0 begins incrementing on that clock. When the
value matches that of compare match constant register 0 (CMCOR0), CMCNT0 is cle
H'0000 and the CMF flag in CMCSR0 is set to 1.

CMCNT0 is initialized to H'0000 by a reset, but retains its previous value in standby r

Bit:	15	14	13	12	11	10	9
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value:	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1
Initial value:	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

11.4.3 Operation

Period Count Operation

When an internal clock is selected with the CKS1 and CKS0 bits in the CMCSR0 regis
STR bit in CMSTR is set to 1, CMCNT0 begins incrementing on the selected clock. W
CMCNT counter value matches that of CMCOR0, the CMCNT0 counter is cleared to H
the CMF flag in the CMCSR0 register is set to 1. The CMCNT0 counter begins countin
from H'0000.

Figure 11.25 shows the compare match counter operation.

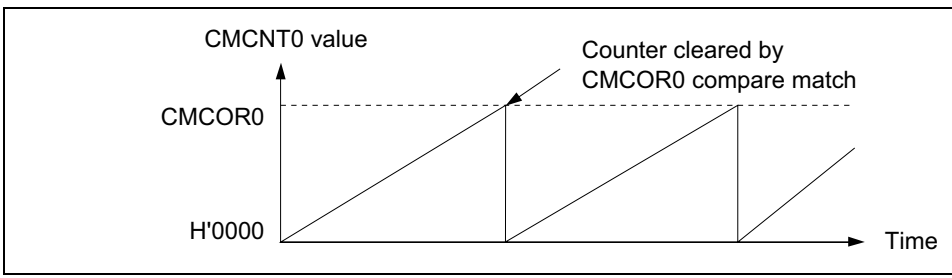


Figure 11.25 Counter Operation

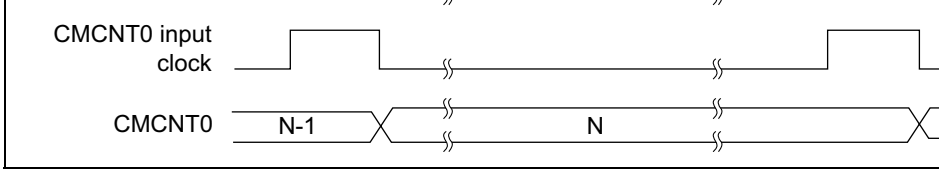


Figure 11.26 Count Timing

11.4.4 Compare Match

Compare Match Flag Setting Timing

The CMF bit in the CMCSR0 register is set to 1 by the compare match signal generate
CMCOR0 register and the CMCNT0 counter match. The compare match signal is gen
final state of the match (timing at which the CMCNT0 counter matching count value i
Consequently, after the CMCOR0 register and the CMCNT0 counter match, a compar
signal will not be generated until a CMCNT0 counter input clock occurs. Figure 11.27
CMF bit setting timing.

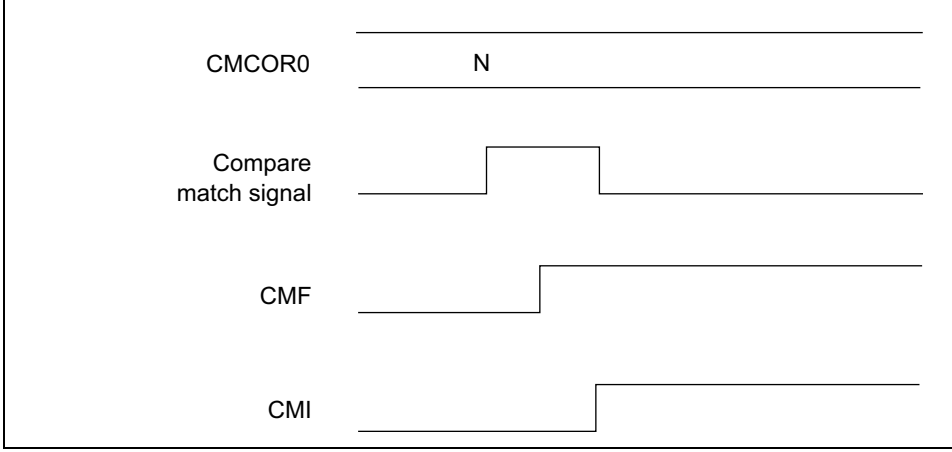


Figure 11.27 CMF Setting Timing

Compare Match Flag Clearing Timing

The CMF bit in the CMCSR0 register is cleared by writing 0 to it after reading 1. Figur
shows the timing when the CMF bit is cleared by the CPU.

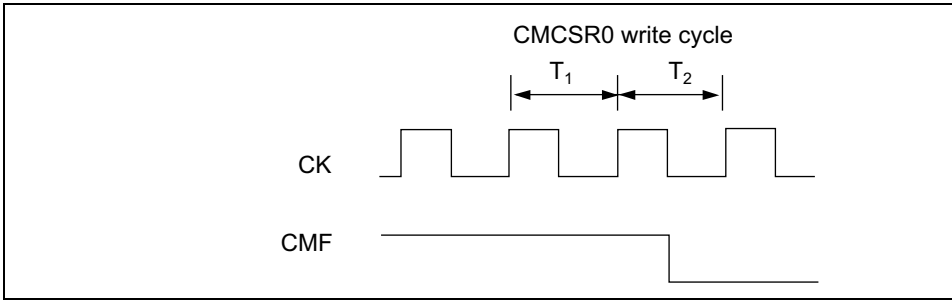


Figure 11.28 Timing of CMF Clearing by the CPU

Table 11.8 Transfer Conditions and Register Settings for Transfer between On- and External Memory

Transfer Conditions	Register	Settin
Transfer source: RDR1 of on-chip IrDA	SAR3	H'040
Transfer destination: External memory	DAR3	H'004
Number of transfers: 64	DMATCR3	H'000
Transfer source address: Fixed	CHCR3	H'000
Transfer destination address: Incremented		
Transfer request source: IrDA (RXI1)		
Bus mode: Cycle-steal		
Transfer unit: Byte		
Interrupt request generated at end of transfer		
Channel priority order: 0 > 2 > 3 > 1	DMAOR	H'010

Transfer source: On-chip A/D converter	SAR2	H'04000
Transfer destination: Internal memory	DAR2	H'00400
Number of transfers: 128 (reloading 32 times)	DMATCR2	H'00000
Transfer source address: Incremented	CHCR2	H'00089
Transfer destination address: Decremented		
Transfer request source: A/D converter		
Bus mode: Burst		
Transfer unit: Longword		
Interrupt request generated at end of transfer		
Channel priority order: 0 > 2 > 3 > 1	DMAOR	H'0101

When the address reload function is on, the value set in SAR returns to the initially set four transfers. In this example, when a transfer request is generated from the A/D conve data is read from the register at address H'04000080 in the A/D converter, and is written external memory address H'00400000. Since longword data has been transferred, the va SAR and DAR are H'04000084 and H'003FFFFC, respectively. The bus is kept and dat are performed successively because this transfer is in burst mode.

After four transfers end, fifth and sixth transfers are performed if the address reload fun and the value in SAR is incremented from H'0400008C to H'04000090, H'04000094… address reload function is on, DMA transfer stops after the fourth transfer ends, and the request signal to the CPU is cleared. At this time, the value stored in SAR is not increm from H'0400008C to H'04000090, but returns to the initially set value, H'04000080. Th DAR continues to be decremented regardless of whether the address reload function is

Bus right	Released	Held
DMAC operation	Stops	Keeps operating
Interrupt	Not generated	Not generated
Transfer request source flag clearing	Executed	Not executed

Notes: 1. An interrupt is generated regardless of whether the address reload function if transfers are executed until the value in DMATCR reaches 0 and the IE b has been set to 1.
2. The transfer request source flag is cleared regardless of whether the addre function is on or off, if transfers are executed until the value in DMATCR re
3. Specify burst mode when using the address reload function. This function r correctly executed in cycle-steal mode.
4. Set a multiple of four in DMATCR when using the address reload function. may not be correctly executed if other values are specified.

11.5.3 Example of DMA Transfer between External Memory and SCIF Trans (Indirect Address On)

In this example, DMA transfer is performed between the external memory specified b address (transfer source) and the SCIF transmitter (transfer destination) using DMAC Table 11.11 shows the transfer conditions and register settings. In addition, the trigger number of transmit FIFO data bytes is set to 1 (TTRG1 = TTRG0 = 1 in SCFCR).

Number of transfers: 16	DMATCR3	H'0000
Transfer source address: Incremented	CHCR3	H'0001
Transfer destination address: Fixed		
Transfer request source: SCIF (TXI2)		
Bus mode: Cycle-steal		
Transfer unit: Byte		
No interrupt request generated at end of transfer		
Channel priority order: 0 > 1 > 2 > 3	DMAOR	H'0001

If the indirect address is on, data stored in the address set in SAR is not used as transfer data. In the indirect address, after the value stored in the address set in SAR is read, that is used as an address again, and the value stored in that address is read and stored in the set in DAR.

In the example shown in table 11.11, when an SCIF transfer request is generated, the D the value in address H'00400000 set in SAR3. Since the value H'00450000 is stored in t address, the DMAC reads the value H'00450000. Next, the DMAC uses that read value address again, and reads the value H'55 stored in that address. Then, the DMAC writes H'55 to address H'04000156 set in DAR3; this completes one indirect address transfer.

In the indirect address, when data is read first from the address set in SAR3, the data tra is always longword regardless of the settings of the TS0 and TS1 bits that specify the tr size. However, whether the transfer source address is fixed, incremented, or decremente specified by the SM0 and SM1 bits. Therefore, in this example, though the transfer data specified as byte, the value in SAR3 is H'00400004 when one transfer ends. Write oper the same as in normal dual address transfer.

4. Before entering standby mode, the DME bit in DMAOR must be cleared to 0 and t accepted by the DMAC completed.
5. The on-chip peripherals which the DMAC can access are the IrDA, SCIF, A/D con converter, and I/O ports. Do not access other peripherals with the DMAC.
6. When starting up the DMAC, set CHCR or DMAOR last. Normal operation is not if settings for another register are made last.
7. Even if the maximum number of transfers are performed in the same channel after DMATCR count reaches 0 and DMA transfer ends normally, write 0 to DMATCR Otherwise, normal DMA transfer may not be performed.
8. When using the address reload function, specify burst mode as the transfer mode. I mode, normal DMA transfer may not be performed.
9. When using the address reload function, set a multiple of four in DMATCR. Norm is not guaranteed if other values are specified.
10. When detecting an external request at the falling edge, keep the external request pi setting the DMAC.
11. Do not access the space from H'4000062 to H'400006F, which is not used in the D Accessing this space may cause malfunctions.
12. The $\overline{\text{WAIT}}$ signal is ignored in the case of a write to external address space in dual mode with 16-byte transfer, or transfer from an external device with DACK to exte space in signal address mode with 16-byte transfer.
13. DMAC transfers should not be performed in the sleep mode under conditions othe the clock ratio of Iϕ (on-chip clock) to Bϕ (bus clock) is 1:1.
14. When the following three conditions are all met, the frequency control register (FR should not be changed while a DMAC transfer is in progress.
    - Bits IFC2 to IFC0 are changed.
    - STC2 to STC0 in FRQCR are not changed.
    - The clock ratio of Iϕ (on-chip clock) to Bϕ (bus clock) after the change is othe

The TMU has the following features:

- Each channel is provided with an auto-reload 32-bit down counter.
- Channel 2 is provided with an input capture function.
- All channels are provided with 32-bit constant registers and 32-bit down counters
  read or written to at any time.
- All channels generate interrupt requests when the 32-bit down counter underflows
  (H'00000000 → H'FFFFFFFF).
- Allows selection between 6 counter input clocks: External clock (TCLK), on-chip
  clock (16 kHz), Pϕ/4, Pϕ/16, Pϕ/64, Pϕ/256. (Pϕ is the internal clock for peripher
  See section 9, On-Chip Oscillation Circuits, for more information on the clock pul
- All channels can operate when the SH7709S is in standby mode: When the RTC o
  is being used as the counter input clock, the SH7709S is still able to count in stand
- Synchronized read: TCNT is a sequentially changing 32-bit register. Since the peri
  module used has an internal bus width of 16 bits, a time lag can occur between the
  the upper 16 bits and lower 16 bits are read. To correct the discrepancy in the cou
  value caused by this time lag, a synchronization circuit is built into the TCNT so t
  32-bit data in the TCNT can be read at once.
- The maximum operating frequency of the 32-bit counter is 2 MHz on all channels:
  SH7709S so that the clock input to the timer counters of each channel (obtained b
  the external clock and internal clock with the prescaler) does not exceed the maxi
  operating frequency.

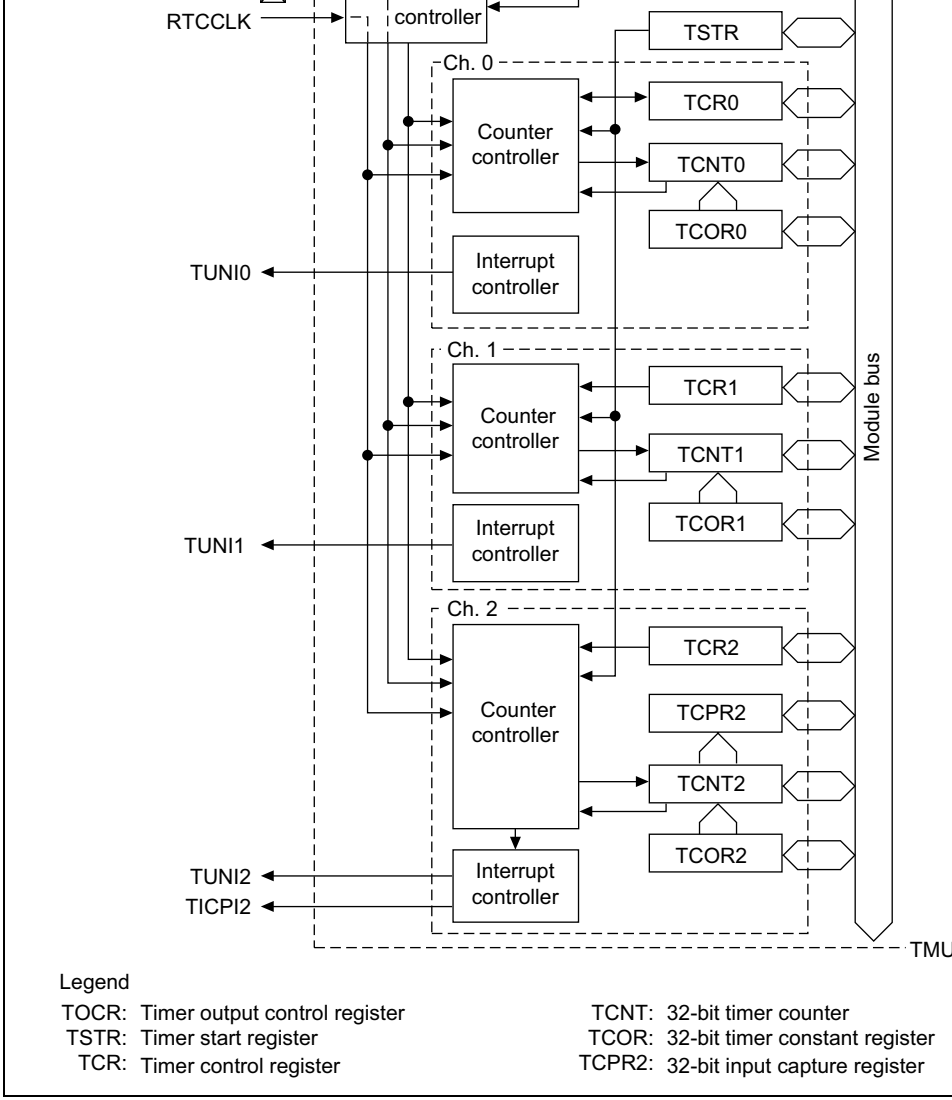


Figure 12.1 Block Diagram of TMU

12.1.4 Register Configuration

Table 12.2 shows the TMU register configuration.

Table 12.2 TMU Registers

Channel	Register	Abbre-viation	R/W	Initial Value*	Address
Common	Timer output control register	TOCR	R/W	H'00	H'FFFFFE9
	Timer start register	TSTR	R/W	H'00	H'FFFFFE9
0	Timer constant register 0	TCOR0	R/W	H'FFFFFFFF	H'FFFFFE9
	Timer counter 0	TCNT0	R/W	H'FFFFFFFF	H'FFFFFE9
	Timer control register 0	TCR0	R/W	H'0000	H'FFFFFE9
1	Timer constant register 1	TCOR1	R/W	H'FFFFFFFF	H'FFFFFEA
	Timer counter 1	TCNT1	R/W	H'FFFFFFFF	H'FFFFFEA
	Timer control register 1	TCR1	R/W	H'0000	H'FFFFFEA
2	Timer constant register 2	TCOR2	R/W	H'FFFFFFFF	H'FFFFFEA
	Timer counter 2	TCNT2	R/W	H'FFFFFFFF	H'FFFFFEB
	Timer control register 2	TCR2	R/W	H'0000	H'FFFFFEB
	Input capture register 2	TCPR2	R	Undefined	H'FFFFFEB

Note: * Initialized by power-on resets or manual resets.

Bit:	7	6	5	4	3	2	1
	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R

Bits 7 to 1—Reserved: These bits are always read as 0. The write value should always

Bit 0—Timer Clock Pin Control (TCOE): Selects use of the timer clock pin (TCLK) external clock output pin or input pin for input capture control for the on-chip timer, or output pin for the on-chip RTC output clock. Since the TCLK pin is multiplexed as the when the pin is used as TCLK, bits PH7MD1 and PH7MD0 in the PHCR register shoul 00 (the "other function" setting).

Bit 0: TCOE	Description
0	Timer clock pin (TCLK) used as external clock input or input capt input pin for the on-chip timer (I
1	Timer clock pin (TCLK) used as output pin for on-chip RTC outpu

12.2.2 Timer Start Register (TSTR)

TSTR is an 8-bit readable/writable register that selects whether to run or halt the timer (TCNT) for channels 0–2. TSTR is initialized to H'00 by a power-on reset or manual re not initialized in standby mode when the input clock selected for the channel is the on-c clock (RTCCLK). Only when an external clock (TCLK) or the peripheral clock (Pϕ) is input clock, it is initialized in standby mode when the multiplication ratio of PLL circui changed or when the MSTP2 bit in STBCR is set to 1.

Bit:	7	6	5	4	3	2	1
	—	—	—	—	—	STR2	STR1
Initial value:	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W

Bit 1: STR1	Description
0	TCNT1 count halted
1	TCNT1 counts

Bit 0—Counter Start 0 (STR0): Selects whether to run or halt timer counter 0 (TCN

Bit 0: STR0	Description
0	TCNT0 count halted
1	TCNT0 counts

12.2.3 Timer Control Registers (TCR)

The timer control registers (TCR) control the timer counters (TCNT) and interrupts. T
three TCR registers, one for each channel.

The TCR registers are 16-bit readable/writable registers that control the issuance of in
when the flag indicating timer counter (TCNT) underflow has been set to 1, and also c
counter clock selection. When the external clock has been selected, they also select its
Additionally, TCR2 controls the channel 2 input capture function and the issuance of i
during input capture. The TCR registers are initialized to H'0000 by a power-on reset
reset, but are not initialized in standby mode and retain their contents.

	—	—	UNIE	CKEG1	CKEG0	TPSC2	TPSC
Initial value:	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R/W	R/W	R/W

Channel 2 TCR Bit Configuration:

Bit:	15	14	13	12	11	10	9
	—	—	—	—	—	—	ICPF
Initial value:	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W

Bit:	7	6	5	4	3	2	1
	ICPE1	ICPE0	UNIE	CKEG1	CKEG0	TPSC2	TPSC1
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bits 15 to 10, 9 (except TCR2), 7, and 6 (except TCR2)—Reserved: These bits are a
as 0. The write value should always be 0.

Bit 9—Input Capture Interrupt Flag (ICPF): A function of channel 2 only: the flag i
input capture is requested via the TCLK pin.

Bit 9: ICPF	Description
0	No input capture request has been issued Clearing condition: When 0 is written to ICPF (I
1	Input capture has been requested via the TCLK pin Setting condition: When input capture is requested via the TCLK pin*

Note: * Contents do not change when 1 is written to ICPF.

Bits 7 and 6—Input Capture Control (ICPE1, ICPE0): A function of channel 2 on
determines whether the input capture function can be used, and when used, whether or
enable interrupts.

When using this input capture function it is necessary to set the TCLK pin to input mo
TCOE bit in the TOCR register. Additionally, use the CKEG bit to designate use of ei
rising or falling edge of the TCLK pin to set the value in TCNT2 in the input capture r
(TCPR2).

Bit 7: ICPE1	Bit 6: ICPE0	Description
0	0	Input capture function is not used
	1	Reserved (Setting prohibited)
1	0	Input capture function is used. Interrupt due to ICF is not enabled
	1	Input capture function is used. Interrupt due to ICF is enabled

Bit 5—Underflow Interrupt Control (UNIE): Controls enabling of interrupt generat
the status flag (UNF) indicating TCNT underflow has been set to 1.

Bit 5: UNIE	Description
0	Interrupt due to UNF (TUNI) is not enabled
1	Interrupt due to UNF (TUNI) is enabled

Bits 2 to 0—Timer Prescaler 2 to 0 (TPSC2 to TPSC0): Select the TCNT count cloc

Bit 2: TPSC2	Bit 1: TPSC1	Bit 0: TPSC0	Description
0	0	0	Internal clock: count on Pϕ/4 (I
		1	Internal clock: count on Pϕ/16
	1	0	Internal clock: count on Pϕ/64
		1	Internal clock: count on Pϕ/256
1	0	0	Internal clock: count on clock output of RTC (RTC CLK)
		1	Count on TCLK pin input
	1	0	Reserved (Setting prohibited)
		1	Reserved (Setting prohibited)

Initial value:	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	23	22	21	20	19	18	17
Initial value:	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9
Initial value:	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1
Initial value:	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

12.2.5 Timer Counters (TCNT)

The timer counters are 32-bit readable/writable registers. The TMU has three timer co for each channel.

TCNT counts down upon input of a clock. The clock input is selected using the TPSC bits in the timer control register (TCR).

When a TCNT count-down results in an underflow (H'00000000 → H'FFFFFFFF), th flag (UNF) in the timer control register (TCR) of the relevant channel is set. The TCO simultaneously set in TCNT itself and the count-down continues from that value.

Bit:	31	30	29	28	27	26	25
Initial value:	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	23	22	21	20	19	18	17
Initial value:	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9
Initial value:	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1
Initial value:	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	31	30	29	28	27	26	25
Initial value:	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R

Bit:	23	22	21	20	19	18	17
Initial value:	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R

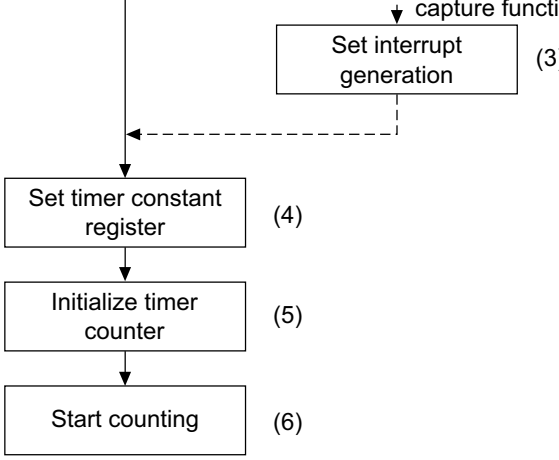
Bit:	15	14	13	12	11	10	9
Initial value:	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R

Bit:	7	6	5	4	3	2	1
Initial value:	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R

counter (TCNT) starts counting. When a TCNT underflows, the UNF flag of the corres
timer control register (TCR) is set. At this time, if the UNIE bit in TCR is 1, an interru
sent to the CPU. Also at this time, the value is copied from TCOR to TCNT and the do
operation is continued.

The count operation is set as follows (figure 12.2):

1. Select the counter clock with the TPSC2–TPSC0 bits in the timer control register (T
external clock is selected, set the TCLK pin to input mode with the TOCE bit in TO
select its edge with the CKEG1 and CKEG0 bits in TCR.
2. Use the UNIE bit in TCR to set whether to generate an interrupt when TCNT under
3. When using the input capture function, set the ICPE bits in TCR, including the choi
whether or not to use the interrupt function (channel 2 only).
4. Set a value in the timer constant register (TCOR) (the cycle is the set value plus 1).
5. Set the initial value in the timer counter (TCNT).
6. Set the STR bit in the timer start register (TSTR) to 1 to start operation.



Note: When an interrupt has been generated, clear the flag in the interrupt handler
caused it. If interrupts are enabled without clearing the flag, another interrupt
generated.

Figure 12.2 Setting the Count Operation

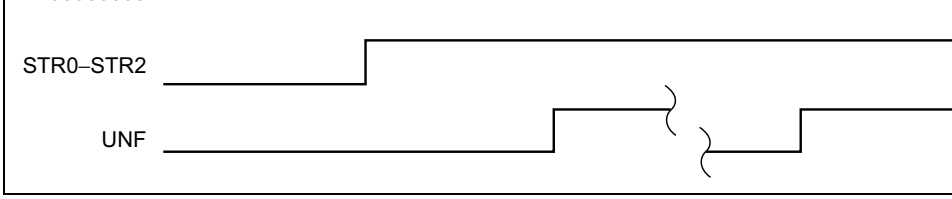


Figure 12.3 Auto-Reload Count Operation

TCNT Count Timing:

- Internal Clock Operation: Set the TPSC2–TPSC0 bits in TCR to select whether peri
  module clock Pϕ or one of the four internal clocks created by dividing it is used (Pϕ
  Pϕ/64, Pϕ/256). Figure 12.4 shows the timing.

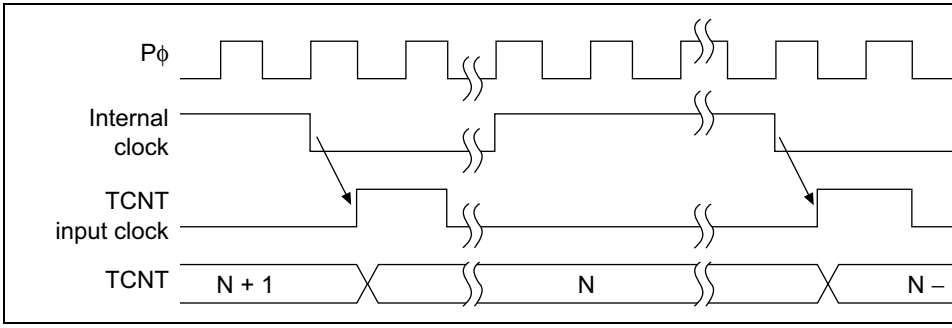


Figure 12.4 Count Timing when Operating on Internal Clock

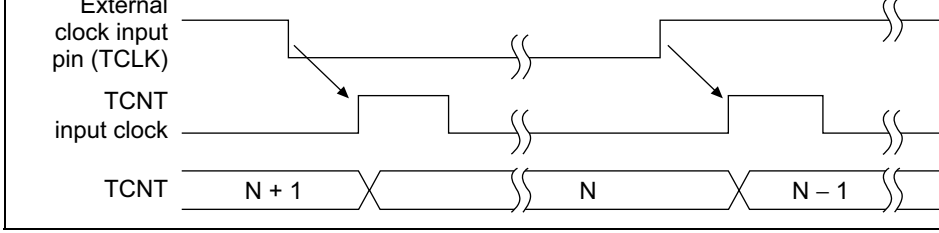


Figure 12.5 Count Timing when Operating on External Clock (Both Edges I

- On-Chip RTC Clock Operation: Set the TPSC2–TPSC0 bits in TCR to select the c clock as the timer clock. Figure 12.6 shows the timing.

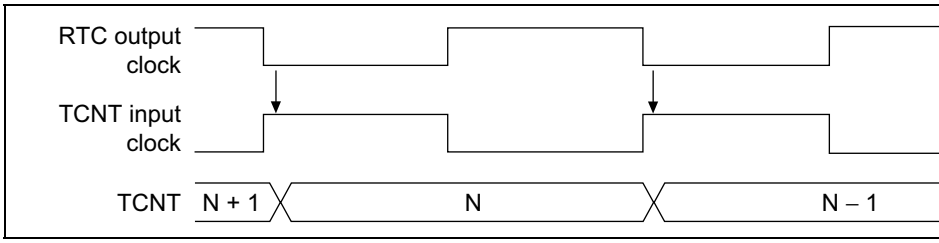


Figure 12.6 Count Timing when Operating on On-Chip RTC Clock

12.3.2 Input Capture Function

Channel 2 has an input capture function (figure 12.7). When using the input capture fu the TCLK pin to input mode with the TCOE bit in the timer output control register (T set the timer operation clock to internal clock or on-chip RTC clock with the TPCS2–T in the timer control register (TCR2). Also, designate use of the input capture function to generate interrupts on input capture with the IPCE1–IPCE0 bits in TCR2, and desig of either the rising or falling edge of the TCLK pin to set the timer counter (TCNT2) v input capture register (TCPR2) with the CKEG1–CKEG0 bits in TCR2.

The input capture function cannot be used in standby mode.

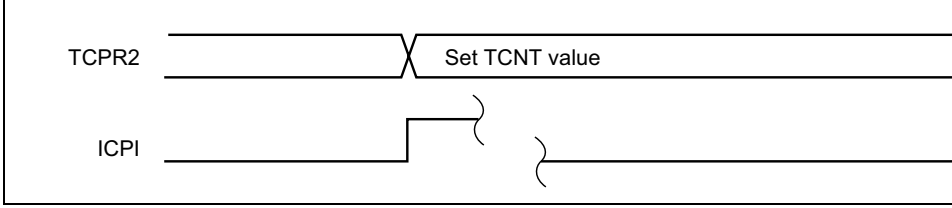


Figure 12.7 Operation Timing when Using Input Capture Function (Using TCLK Rising Edge)

12.4 Interrupts

There are two sources of TMU interrupts: underflow interrupts (TUNI) and interrupts v
the input capture function (TICPI2).

12.4.1 Status Flag Setting Timing

UNF is set to 1 when the TCNT underflows. Figure 12.8 shows the timing.

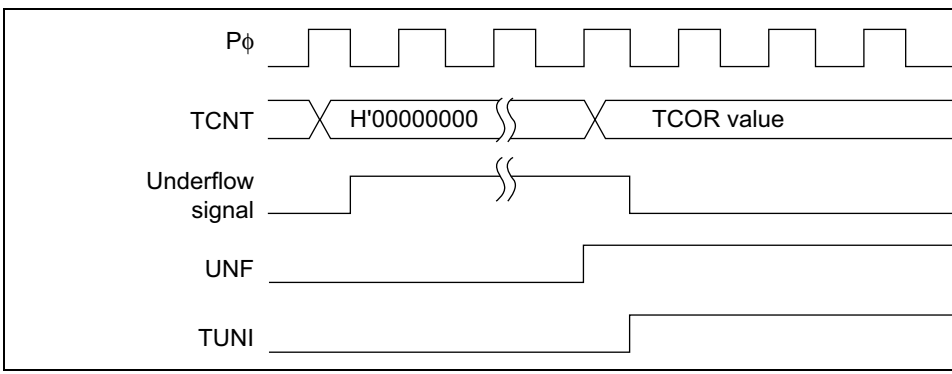


Figure 12.8 UNF Setting Timing

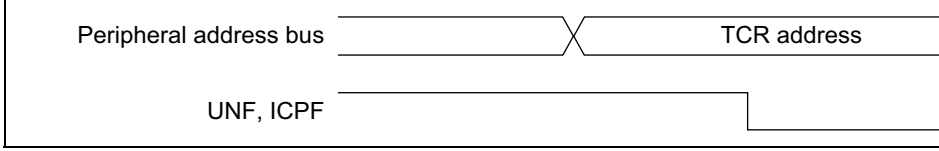


Figure 12.9 Status Flag Clearing Timing

12.4.3 Interrupt Sources and Priorities

The TMU produces underflow interrupts for each channel. When the interrupt request interrupt enable bit are both set to 1, an interrupt is requested. Codes are set in the inte registers (INTEVT, INTEVT2) for these interrupts and interrupt handling occurs acco codes.

The relative priorities of channels can be changed using the interrupt controller (see se Exception Handling, and section 6, Interrupt Controller (INTC)). Table 12.3 lists TMU sources.

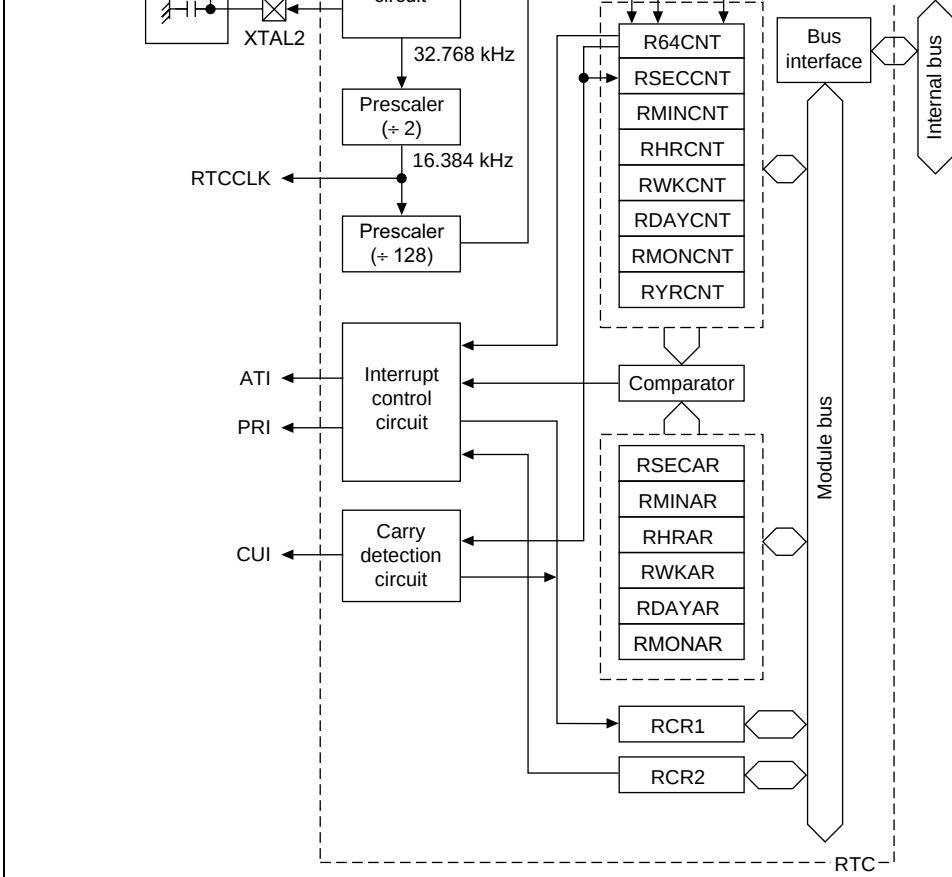
Table 12.3 TMU Interrupt Sources

Channel	Interrupt Source	Description	Priority
0	TUNI0	Underflow interrupt 0	High
1	TUNI1	Underflow interrupt 1	↕
2	TUNI2	Underflow interrupt 2	↕
2	TICPI2	Input capture interrupt 2	Low

12.5.2 Reading Registers

Synchronization processing is performed for timer counting during register reads. Whe
counting and register read processing are performed simultaneously, the register value
TCNT counting down (with synchronization processing) is read.

- Clock and calendar functions (BCD display): Seconds, minutes, hours, date, day o month, and year
- 1-Hz to 64-Hz timer (binary display)
- Start/stop function
- 30-second adjust function
- Alarm interrupt: Frame comparison of seconds, minutes, hours, date, day of the we month can be used as conditions for the alarm interrupt
- Cyclic interrupts: The interrupt cycle may be 1/256 second, 1/64 second, 1/16 seco second, 1/2 second, 1 second, or 2 seconds
- Carry interrupt: A carry interrupt indicates when a carry occurs during a counter re
- Automatic leap year correction



Legend

R64CNT:	64 Hz counter	RSECAR:	Second alarm register
RSECCNT:	Second counter	RMINAR:	Minute alarm register
RMINCNT:	Minute counter	RHRAR:	Hour alarm register
RHRCNT:	Hour counter	RWKAR:	Day-of-week alarm register
RWKCNT:	Day-of-week counter	RDAYAR:	Day alarm register
RDAYCNT:	Day counter	RMONAR:	Month alarm register
RMONCNT:	Month counter	RCR1:	RTC control register 1
RYRCNT:	Year counter	RCR2:	RTC control register 2

Figure 13.1 Block Diagram of RTC

Clock input/clock output	TCLK	I/O	External clock input pin/inp control input pin/realtime c output pin (shared by TMU
Dedicated power-supply pin for RTC	Vcc–RTC	—	Dedicated power-supply p
Dedicated GND pin for RTC	Vss–RTC	—	Dedicated GND pin for RT

Notes: 1. Except in hardware standby mode, power must be supplied to all power su
including these, even when only the RTC is used (including standby mode)
2. When the RTC is not used, pull EXTAL2 up (to Vcc) and make no connecti
XTAL2.

Minute counter	RMINCNT	R/W	Undefined	H'FFFFFEC4	8
Hour counter	RHRCNT	R/W	Undefined	H'FFFFFEC6	8
Day of week counter	RWKCNT	R/W	Undefined	H'FFFFFEC8	8
Date counter	RDAYCNT	R/W	Undefined	H'FFFFFECA	8
Month counter	RMONCNT	R/W	Undefined	H'FFFFFECC	8
Year counter	RYRCNT	R/W	Undefined	H'FFFFFECE	8
Second alarm register	RSECAR	R/W	Undefined*	H'FFFFFED0	8
Minute alarm register	RMINAR	R/W	Undefined*	H'FFFFFED2	8
Hour alarm register	RHRAR	R/W	Undefined*	H'FFFFFED4	8
Day of week alarm register	RWKAR	R/W	Undefined*	H'FFFFFED6	8
Date alarm register	RDAYAR	R/W	Undefined*	H'FFFFFED8	8
Month alarm register	RMONAR	R/W	Undefined*	H'FFFFFEDA	8
RTC control register 1	RCR1	R/W	H'00	H'FFFFFEDC	8
RTC control register 2	RCR2	R/W	H'09	H'FFFFFEDE	8

Note: * Only the ENB bits of each register are initialized.

R64CNT is not initialized by a power-on reset or manual reset, or in standby mode.

Bit 7 is always read as 0.

Bit:	7	6	5	4	3	2	1
	—	1Hz	2Hz	4Hz	8Hz	16Hz	32H
Initial value:	0	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R

13.2.2 Second Counter (RSECCNT)

The second counter (RSECCNT) is an 8-bit readable/writable register used for setting
the BCD-coded second section of the RTC. The count operation is performed by a car
second of the 64-Hz counter.

The range that can be set is 00–59 (decimal). Errant operation will result if any other v
Carry out write processing after halting the count operation with the START bit in RC

RSECCNT is not initialized by a power-on reset or manual reset, or in standby mode.

Bit:	7	6	5	4	3	2	1
	—	10 seconds			1 second		
Initial value:	0	—	—	—	—	—	—
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W

RMINCNT is not initialized by a power-on reset or manual reset, or in standby mode.

Bit:	7	6	5	4	3	2	1
	—	10 minutes			1 minute		
Initial value:	0	—	—	—	—	—	—
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W

13.2.4 Hour Counter (RHRCNT)

The hour counter (RHRCNT) is an 8-bit readable/writable register used for setting/coun
BCD-coded hour section of the RTC. The count operation is performed by a carry for e
of the minute counter.

The range that can be set is 00–23 (decimal). Errant operation will result if any other va
Carry out write processing after halting the count operation with the START bit in RCR
a carry flag as shown in figure 13.2.

RHRCNT is not initialized by a power-on reset or manual reset, or in standby mode.

Bit:	7	6	5	4	3	2	1
	—	—	10 hours			1 hour	
Initial value:	0	0	—	—	—	—	—
R/W:	R	R	R/W	R/W	R/W	R/W	R/W

RWKCNT is not initialized by a power-on reset or manual reset, or in standby mode.

Bit:	7	6	5	4	3	2	1
	—	—	—	—	—	Day of w	
Initial value:	0	0	0	0	0	—	—
R/W:	R	R	R	R	R	R/W	R/W

Days of the week are coded as shown in table 13.3.

Table 13.3 Day-of-Week Codes (RWKCNT)

Day of Week	Code
Sunday	0
Monday	1
Tuesday	2
Wednesday	3
Thursday	4
Friday	5
Saturday	6

RDAYCNT is not initialized by a power-on reset or manual reset, or in standby mode.

The RDAYCNT range that can be set changes with each month and in leap years. Pleas
the correct setting.

Bit:	7	6	5	4	3	2	1
	—	—	10 days			1 day	
Initial value:	0	0	—	—	—	—	—
R/W:	R	R	R/W	R/W	R/W	R/W	R/W

13.2.7 Month Counter (RMONCNT)

The month counter (RMONCNT) is an 8-bit readable/writable register used for setting/
the BCD-coded month section of the RTC. The count operation is performed by a carry
month of the date counter.

The range that can be set is 00–12 (decimal). Errant operation will result if any other va
Carry out write processing after halting the count operation with the START bit in RCR

RMONCNT is not initialized by a power-on reset or manual reset, or in standby mode.

Bit:	7	6	5	4	3	2	1
	—	—	—	10 months		1 month	
Initial value:	0	0	0	—	—	—	—
R/W:	R	R	R	R/W	R/W	R/W	R/W

RTCCNT is not initialized by a power-on reset or manual reset, or in standby mode.

Leap years are recognized by dividing the year counter value by 4 and obtaining a frac of 0. The year counter value: 00 is included in leap years.

Bit:	7	6	5	4	3	2	1
	10 years				1 year		
Initial value:	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

13.2.9 Second Alarm Register (RSECAR)

The second alarm register (RSECAR) is an 8-bit readable/writable register, and an ala corresponding to the BCD-coded second section counter RSECCNT of the RTC. Whe bit is set to 1, a comparison with the RSECCNT value is performed. From among the RSECAR/RMINAR/RHRAR/RWKAR/RDAYAR/RMONAR registers, the counter a register comparison is performed only on those with ENB bits set to 1, and if each of t coincide, an RTC alarm interrupt is generated.

The range that can be set is 00–59 (decimal) + ENB bit. Errant operation will result if value is set.

The ENB bit in RSECAR is initialized to 0 by a power-on reset. The remaining RSEC are not initialized and retain their contents by a manual reset, or in standby mode.

Bit:	7	6	5	4	3	2	1
	ENB	10 seconds				1 second	
Initial value:	0	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The range that can be set is 00–59 (decimal) + ENB bit. Errant operation will result if a value is set.

The ENB bit in RMINAR is initialized by a power-on reset. The remaining RMINAR f not initialized and retain their contents by a manual reset, or in standby mode.

Bit:	7	6	5	4	3	2	1
	ENB	10 minutes			1 minute		
Initial value:	0	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

13.2.11 Hour Alarm Register (RHRAR)

The hour alarm register (RHRAR) is an 8-bit readable/writable register, and an alarm re corresponding to the BCD-coded hour section counter RHRCNT of the RTC. When the set to 1, a comparison with the RHRCNT value is performed. From among the RSECAR/RMINAR/RHRAR/RWKAR/RDAYAR/RMONAR registers, the counter an register comparison is performed only on those with ENB bits set to 1, and if each of th coincide, an RTC alarm interrupt is generated.

The range that can be set is 00–23 (decimal) + ENB bit. Errant operation will result if a value is set.

The ENB bit in RHRAR is initialized by a power-on reset. The remaining RHRAR fiel initialized and retain their contents by a manual reset, or in standby mode.

Bit:	7	6	5	4	3	2	1
	ENB	—	10 hours		1 hour		
Initial value:	0	0	—	—	—	—	—
R/W:	R/W	R	R/W	R/W	R/W	R/W	R/W

The range that can be set is 0–6 (decimal) + ENB bit. Errant operation will result if an
value is set.

The ENB bit in RWKAR is initialized by a power-on reset. The remaining RWKAR f
initialized and retain their contents by a manual reset, or in standby mode.

Bit:	7	6	5	4	3	2	1
	ENB	—	—	—	—	Day of v	
Initial value:	0	0	0	0	0	—	—
R/W:	R/W	R	R	R	R	R/W	R/W

Days of the week are coded as shown in table 13.4.

Table 13.4 Day-of-Week Codes (RWKAR)

Day of Week	Code
Sunday	0
Monday	1
Tuesday	2
Wednesday	3
Thursday	4
Friday	5
Saturday	6

The range that can be set is 01–31 (decimal) + ENB bit. Errant operation will result if a value is set. The RDAYCNT range that can be set changes with some months and in lea Please confirm the correct setting.

The ENB bit in RDAYAR is initialized by a power-on reset. The remaining RDAYAR not initialized and retain their contents by a manual reset, or in standby mode.

Bit:	7	6	5	4	3	2	1
	ENB	—	10 days			1 day	
Initial value:	0	0	—	—	—	—	—
R/W:	R/W	R	R/W	R/W	R/W	R/W	R/W

13.2.14 Month Alarm Register (RMONAR)

The month alarm register (RMONAR) is an 8-bit readable/writable register, and an alar corresponding to the BCD-coded month section counter RMONCNT of the RTC. Wher bit is set to 1, a comparison with the RMONCNT value is performed. From among the RSECAR, RMINAR, RHRAR, RWKAR, RDAYAR, RMONAR, the counter and alarm comparison is performed only on those with ENB bits set to 1, and if each of those coin RTC alarm interrupt is generated.

The range that can be set is 01–12 (decimal) + ENB bit. Errant operation will result if a value is set.

The ENB bit in RMONAR is initialized by a power-on reset. The remaining RMONAR not initialized and retain their contents by a manual reset, or in standby mode.

Bit:	7	6	5	4	3	2	1
	ENB	—	—	10 months		1 month	
Initial value:	0	0	0	—	—	—	—
R/W:	R/W	R	R	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1
	CF	—	—	CIE	AIE	—	—
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R	R	R/W	R/W	R	R

Bit 7—Carry Flag (CF): Status flag that indicates that a carry has occurred. Setting C indicates reading of a counter register value has occurred when (1) the second counter (2) the 64-Hz counter is carried. A count register value read at this time cannot be gua another read is required.

Bit 7: CF	Description
0	No count up of R64CNT or RSECCNT Clearing condition: When 0 is written to CF
1	Count up of R64CNT or RSECCNT Setting condition: When 1 is written to CF

Bits 6, 5, 2, and 1—Reserved: These bits are always read as 0. The write value shoul 0.

Bit 4—Carry Interrupt Enable Flag (CIE): When the carry flag (CF) is set to 1, the enables interrupts.

Bit 4: CIE	Description
0	A carry interrupt is not generated when the CF flag is set to 1
1	A carry interrupt is generated when the CF flag is set to 1

Bit 0—Alarm Flag (AF): The AF flag is set to 1 when the alarm time set in an alarm r
(only registers with ENB bit set to 1) matches the clock and calendar time. This flag is
0 when 0 is written, but holds its previous value when 1 is written.

Bit 0: AF	Description
0	Clock/calendar and alarm register have not matched since last re Clearing condition: When 0 is written to AF (In
1	Setting condition: Clock/calendar and alarm register have matche registers with ENB set)*

Note: * Contents do not change when 1 is written to AF.

13.2.16 RTC Control Register 2 (RCR2)

The RTC control register 2 (RCR2) is an 8-bit readable/writable register for periodic in
control, 30-second adjustment ADJ, divider circuit RESET, and RTC count start/stop c
initialized to H'09 by a power-on reset. It is initialized except for RTCEN and START
manual reset. It is not initialized, and retains its contents, in standby mode.

Bit:	7	6	5	4	3	2	1
	PEF	PES2	PES1	PES0	RTCEN	ADJ	RESE
Initial value:	0	0	0	0	1	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7—Periodic Interrupt Flag (PEF): Indicates interrupt generation with the period
by the PES bits. When set to 1, PEF generates periodic interrupts.

Bit 7: PEF	Description
0	Interrupts not generated with the period designated by the PES b Clearing condition: When 0 is written to PEF (I
1	Interrupts generated with the period designated by the PES bits Setting condition: When 1 is written to PEF

	1	Periodic interrupt generated every 1/2 s
1	0	Periodic interrupt generated every 1 sec
	1	Periodic interrupt generated every 2 sec

Bit 3—RTCEN: Controls the operation of the crystal oscillator for the RTC.

Bit 3: RTCEN	Description
0	Crystal oscillator for RTC is halted
1	Crystal oscillator for RTC runs

Bit 2—30 Second Adjustment (ADJ): When 1 is written to the ADJ bit, times of 29 less will be rounded to 00 seconds and 30 seconds or more to 1 minute. The divider ci prescaler, and R64CNT will be simultaneously reset. This bit is always read as 0. The duration between when the ADJ bit is set to 1 and when the new setting is reflected in value from the seconds counter (RSECCNT) is approximately 91.6 µs (when a 32.768 oscillator is connected to the EXTAL2 pin).

Bit 2: ADJ	Description
0	Runs normally
1 (Write)	30-second adjustment

Bit 1—Reset (RESET): When 1 is written, initializes the divider circuit (RTC presca R64CNT). This bit is always read as 0.

Bit 1: RESET	Description
0	Runs normally
1 (Write)	Divider circuit is reset

13.3 RTC Operation

13.3.1 Initial Settings of Registers after Power-On

All the registers should be set after the power is turned on.

13.3.2 Setting the Time

Figure 13.2 shows how to set the time when the clock is stopped. This works when the calendar or clock is to be set. Programming can be easily performed.

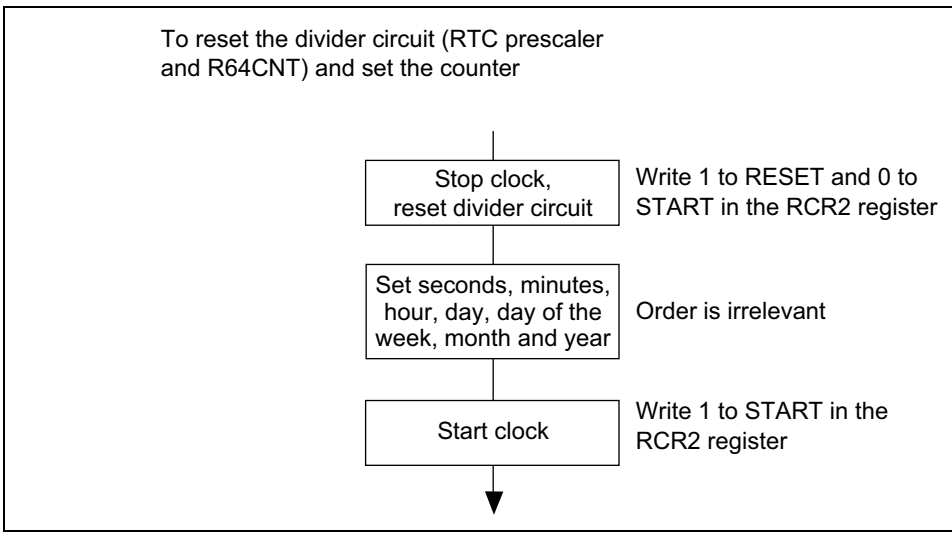


Figure 13.2 Setting the Time

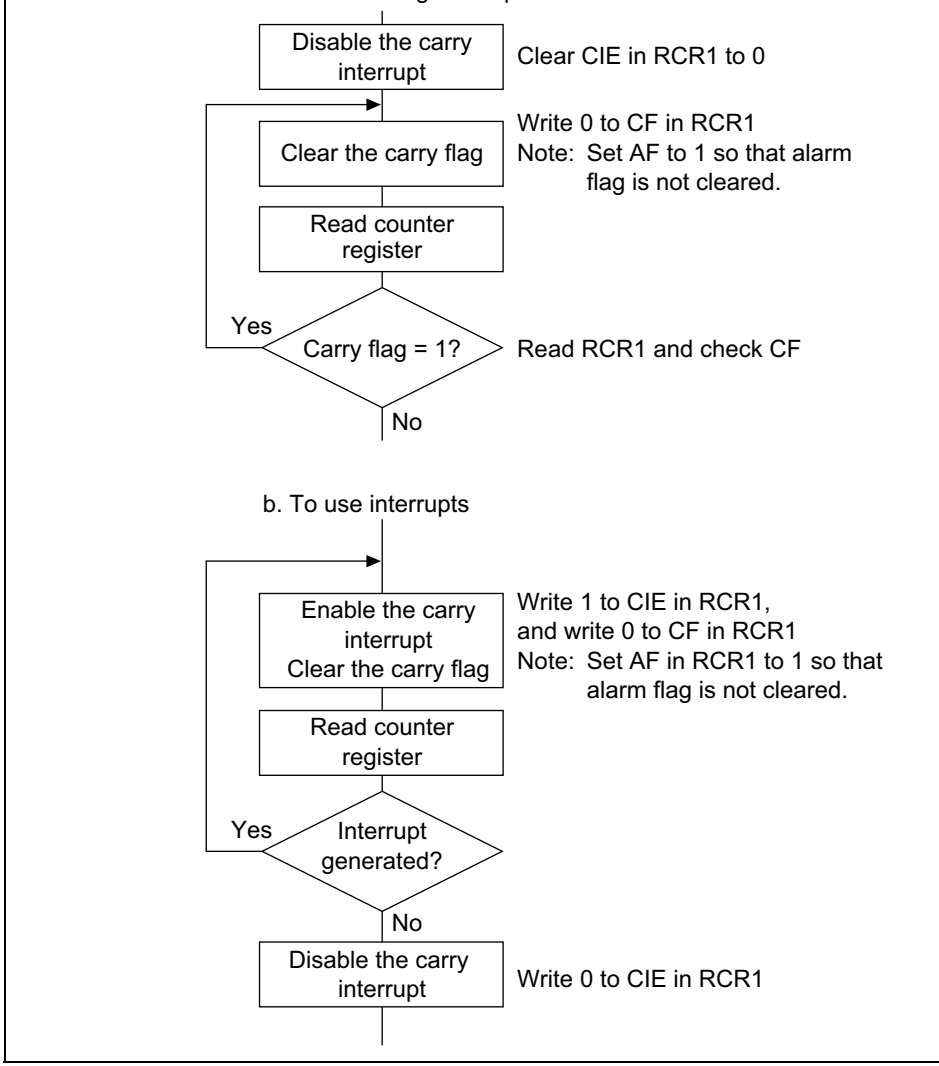


Figure 13.3 Reading the Time

When the clock and alarm times match, 1 is set in the AF bit (bit 0) in RCR1. Alarm de
be checked by reading this bit, but normally it is done by interrupt. If 1 is placed in the
3) in RCR1, an interrupt is generated when an alarm occurs.

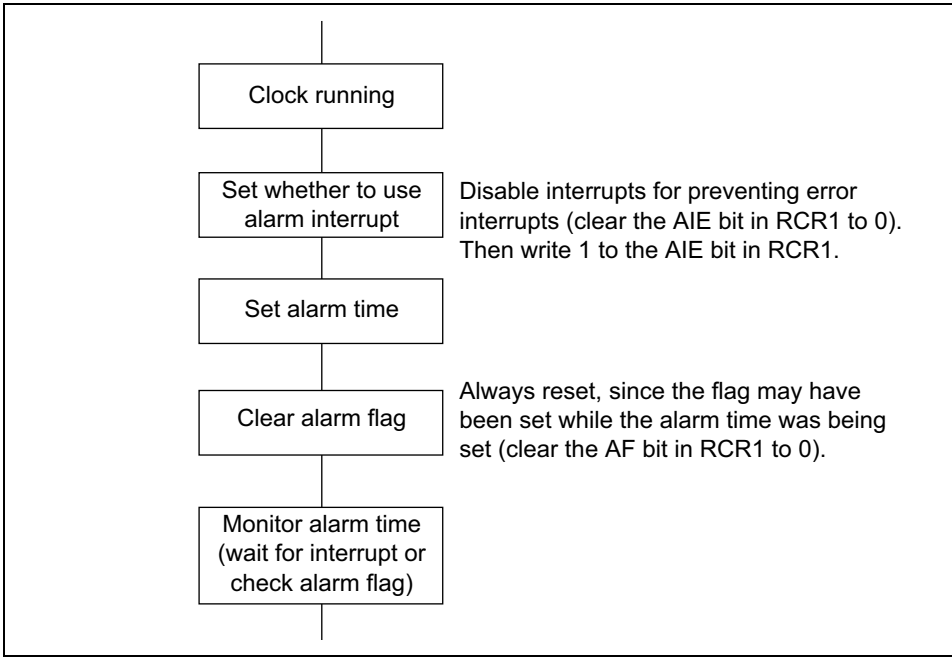
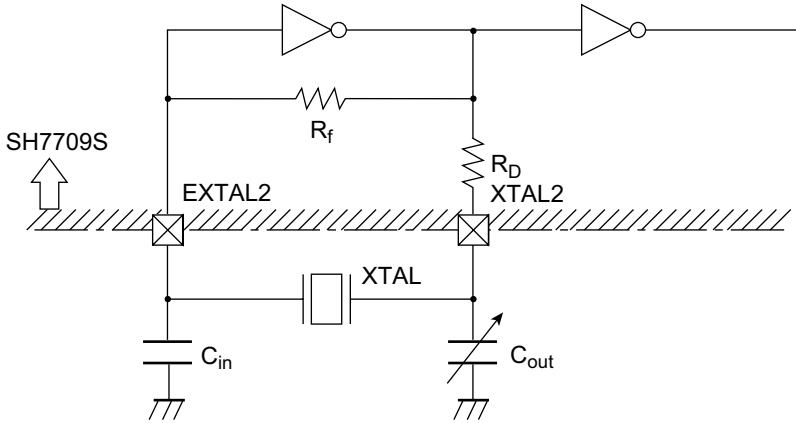


Figure 13.4 Using the Alarm Function



Notes: 1. Select either the C_{in} or C_{out} side for the frequency adjustment variable c
according to requirements such as frequency range, degree of stability,
2. Built-in resistance value R_f (Typ value) = 10 MΩ, R_D (Typ value) = 400 k
3. C_{in} and C_{out} values include floating capacitance due to the wiring. Take
when using a ground plane.
4. The crystal oscillation settling time depends on the mounted circuit cons
floating capacitance, etc., and should be decided after consultation with
crystal resonator manufacturer.
5. Place the crystal resonator and load capacitors C_{in} and C_{out} as close as
to the chip.
(Correct oscillation may not be possible if there is externally induced noi
EXTAL2 and XTAL2 pins.)
6. Ensure that the crystal resonator connection pin (EXTAL2, XTAL2) wirin
routed as far away as possible from other power lines (except GND) and
lines.

Figure 13.5 Example of Crystal Oscillator Circuit Connection

13.4.2 Use of Realtime Clock (RTC) Periodic Interrupts

The method of using the periodic interrupt function is shown in figure 13.6.

A periodic interrupt can be generated periodically at the interval set by the periodic inte
enable flag (PES) in RTC control register 2 (RCR2). When the time set by the periodic
enable flag (PES) has elapsed, the periodic interrupt flag (PEF) is set to 1.

The periodic interrupt flag (PEF) is cleared to 0 upon periodic interrupt generation whe
periodic interrupt enable flag (PES) is set. Periodic interrupt generation can be confirm
reading this bit, but normally the interrupt function is used.

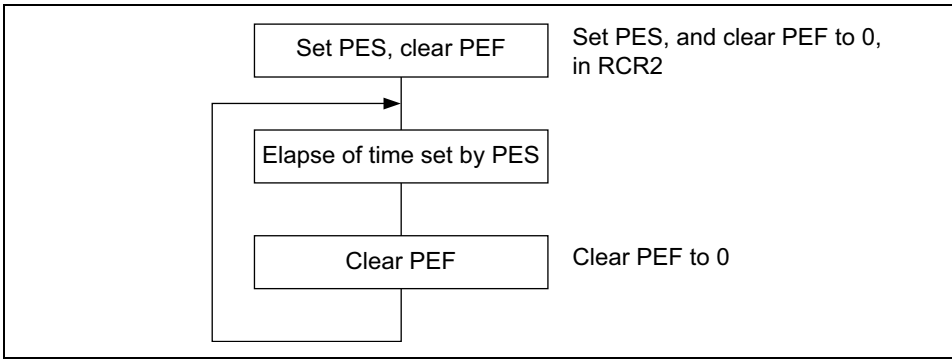


Figure 13.6 Using Periodic Interrupt Function

13.4.3 Precautions when Using RTC Module Standby

Before switching the RTC to module standby, access at least one among the registers R
and TMU.

conforms to the ISO/IEC standard 7816-3 for identification cards. See section 15, Sma
Interface, for more information.

14.1.1 Features

Selection of asynchronous or synchronous as the serial communication mode.

- Asynchronous mode:
  - Serial data communication is synchronized by start-stop in character units. The
    communicate with a universal asynchronous receiver/transmitter (UART), an a
    communication interface adapter (ACIA), or any other communications chip th
    a standard asynchronous serial system. It can also communicate with two or mo
    processors using the multiprocessor communication function. There are 12 sele
    data communication formats.
  - Data length: 7 or 8 bits
  - Stop bit length: 1 or 2 bits
  - Parity: Even, odd, or none
  - Multiprocessor bit: 1 or 0
  - Receive error detection: Parity, overrun, and framing errors
  - Break detection: By reading the RxD level directly from the SC port data regis
    when a framing error occurs
- Synchronous mode:
  - Serial data communication is synchronized with a clock signal. The SCI can co
    with other chips having a synchronous communication function. There is one s
    communication format.
  - Data length: 8 bits
  - Receive error detection: Overrun errors
- Full duplex communication: The transmitting and receiving sections are independe
  SCI can transmit and receive simultaneously. Both sections use double buffering, s
  continuous data transfer is possible in both the transmit and receive directions.
- On-chip baud rate generator with selectable bit rates

Figure 14.1 shows a block diagram of the SCI.

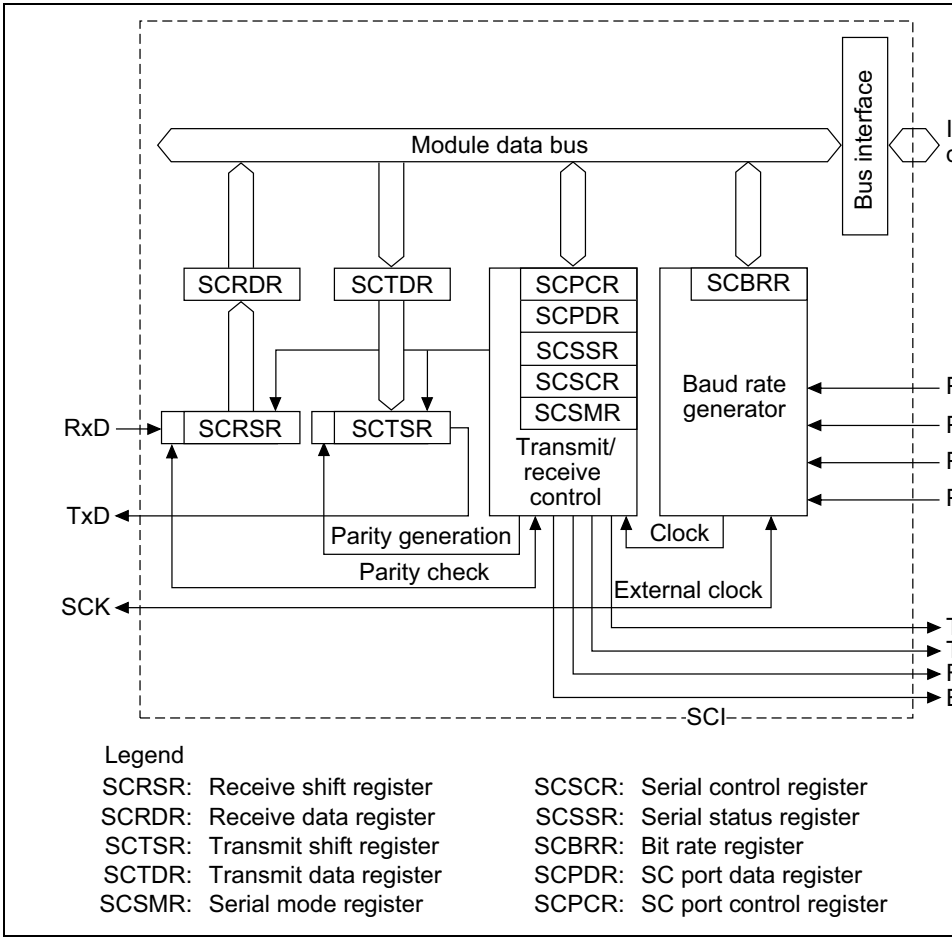
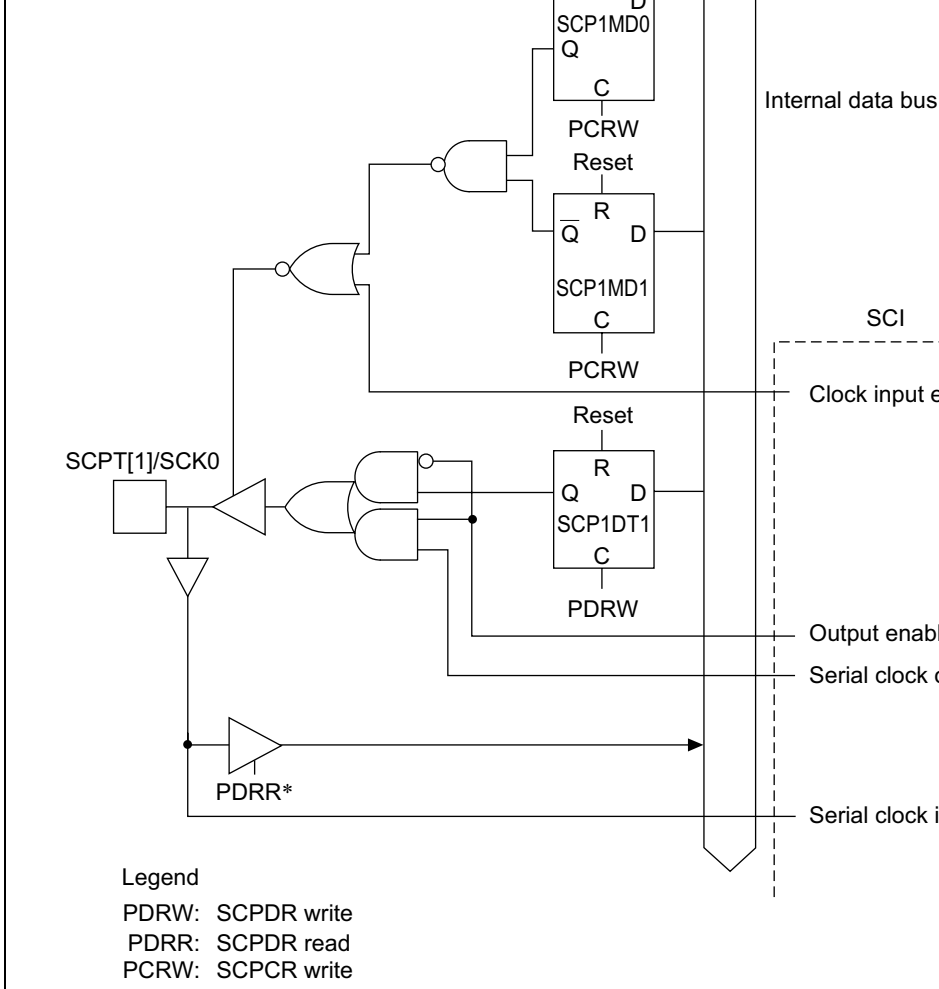


Figure 14.1 Block Diagram of SCI



Note: * When reading the SCK0 pin, clear the C/$\overline{A}$ bit in SCSMR and the CKE1 a
bits in SCSCR to 0, and set the SCP1MD1 bit in SCPCR to 1 (see section
SC Port Control Register (SCPCR)/SC Port Data Register (SCPDR)).

Figure 14.2 SCPT[1]/SCK0 Pin

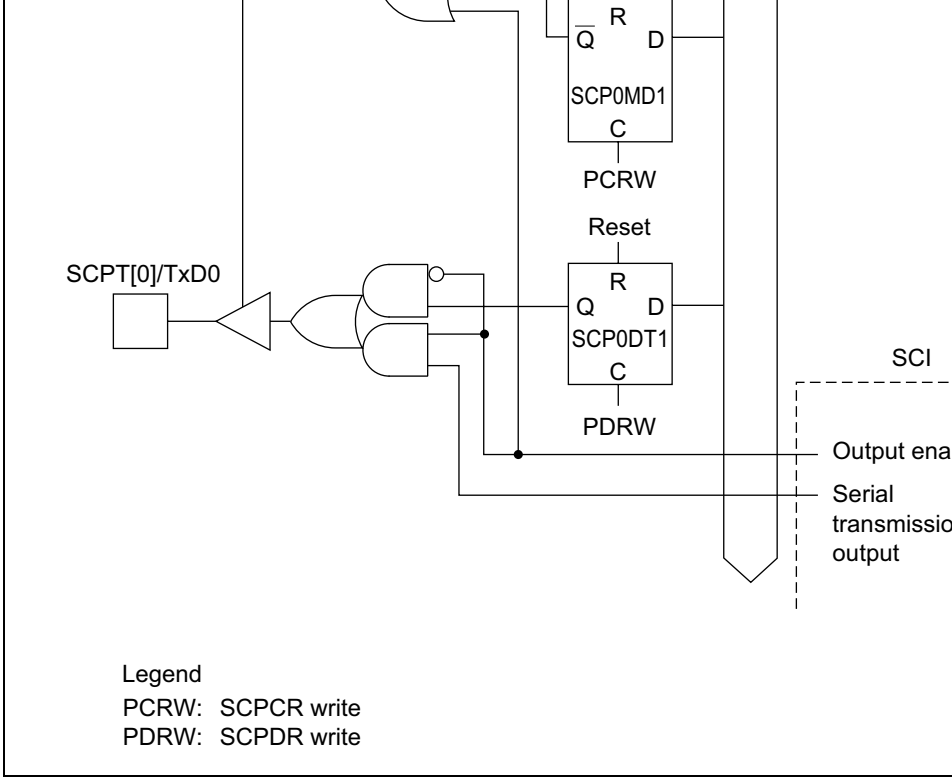
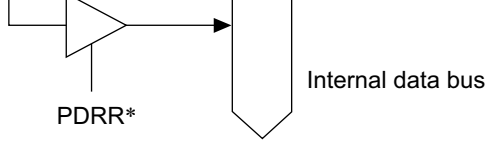


Figure 14.3 SCPT[0]/TxD0 Pin



Legend
PDRR: PDR read

Note: * When reading the RxD0 pin, set the RE bit in SCSCR to 1.

Figure 14.4 SCPT[0]/RxD0 Pin

14.1.3 Pin Configuration

The SCI has the serial pins summarized in table 14.1.

Table 14.1 SCI Pins

Pin Name	Abbreviation	I/O	Function
Serial clock pin	SCK0	I/O	Clock I/O
Receive data pin	RxD0	Input	Receive data input
Transmit data pin	TxD0	Output	Transmit data output

Note: These pins are made to function as serial pins by performing SCI operation set
TE, RE, CKEI, and CKE0 bits in SCSCR and the C/$\overline{A}$ bit in SCSMR. Break stat
transmission and detection can be performed by means of the SCI's SCSPTR r

Bit rate register	SCBRR	R/W	H'FF	H'FFFFFE82	8
Serial control register	SCSCR	R/W	H'00	H'FFFFFE84	8
Transmit data register	SCTDR	R/W	H'FF	H'FFFFFE86	8
Serial status register	SCSSR	R/(W)*	H'84	H'FFFFFE88	8
Receive data register	SCRDR	R	H'00	H'FFFFFE8A	8
SC port data register	SCPDR	R/W	H'00	H'04000136 (H'A4000136)*2	8
SC port control register	SCPCR	R/W	H'A888	H'04000116 (H'A4000116)*2	1(

Notes: These registers are located in area 1 of physical space. Therefore, when the ca either access these registers from the P2 area of logical space or else make an setting using the MMU so that these registers are not cached.

1. The only value that can be written is 0 to clear the flags.
2. When address translation by the MMU does not apply, the address in parent should be used.

14.2 Register Descriptions

14.2.1 Receive Shift Register (SCRSR)

The receive shift register (SCRSR) receives serial data. Data input at the RxD pin is loa SCRSR in the order received, LSB (bit 0) first, converting the data to parallel form. Wh byte has been received, it is automatically transferred to SCRDR. The CPU cannot read SCRSR directly.

Bit:	7	6	5	4	3	2	1
R/W:	—	—	—	—	—	—	—

Bit:	7	6	5	4	3	2	1
Initial value:	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R

14.2.3 Transmit Shift Register (SCTSR)

The transmit shift register (SCTSR) transmits serial data. The SCI loads transmit data
transmit data register (SCTDR) into SCTSR, then transmits the data serially from the
LSB (bit 0) first. After transmitting one-byte data, the SCI automatically loads the nex
data from SCTDR into SCTSR and starts transmitting again. If the TDRE bit in SCSS
however, the SCI does not load the SCTDR contents into SCTSR. The CPU cannot re
to SCTSR directly.

Bit:	7	6	5	4	3	2	1
R/W:	—	—	—	—	—	—	—

Bit:	7	6	5	4	3	2	1
Initial value:	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

14.2.5 Serial Mode Register (SCSMR)

The serial mode register (SCSMR) is an 8-bit register that specifies the SCI serial comm format and selects the clock source for the baud rate generator.

The CPU can always read and write to SCSMR. SCSMR is initialized to H'00 by a rese standby or module standby mode.

Bit:	7	6	5	4	3	2	1
	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	MP	CKS1
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7—Communication Mode (C/$\overline{A}$): Selects whether the SCI operates in asynchrono synchronous mode.

Bit 7: C/$\overline{A}$	Description	
0	Asynchronous mode	(In
1	Synchronous mode	

Bit 5—Parity Enable (PE): Selects whether to add a parity bit to transmit data and to
parity of receive data, in asynchronous mode. In synchronous mode, a parity bit is neit
nor checked, regardless of the PE setting.

Bit 5: PE	Description	
0	Parity bit not added or checked	(
1	Parity bit added and checked*	

Note: * When PE is set to 1, an even or odd parity bit is added to transmit data, deper
parity mode (O/$\overline{E}$) setting. Receive data parity is checked according to the eve
mode setting.

Bit 4—Parity Mode (O/$\overline{E}$): Selects even or odd parity when parity bits are added and
The O/$\overline{E}$ setting is used only in asynchronous mode and only when the parity enable b
to 1 to enable parity addition and checking. The O/$\overline{E}$ setting is ignored in synchronous
asynchronous mode when parity addition and checking is disabled.

Bit 4: O/$\overline{E}$	Description	
0	Even parity*1	(
1	Odd parity*2	

Notes: 1. If even parity is selected, the parity bit is added to transmit data to make an
number of 1s in the transmitted character and parity bit combined. Receive
checked to see if it has an even number of 1s in the received character and
combined.

2. If odd parity is selected, the parity bit is added to transmit data to make an
of 1s in the transmitted character and parity bit combined. Receive data is c
see if it has an odd number of 1s in the received character and parity bit co

0	One stop bit[*1]	(In
1	Two stop bits[*2]	

Notes: 1. When transmitting, a single 1-bit is added at the end of each transmitted cha
2. When transmitting, two 1-bits are added at the end of each transmitted chara

Bit 2—Multiprocessor Mode (MP): Selects multiprocessor format. When multiproces is selected, settings of the parity enable (PE) and parity mode (O/$\overline{E}$) bits are ignored. Th setting is used only in asynchronous mode; it is ignored in synchronous mode. For the multiprocessor communication function, see section 14.3.3, Multiprocessor Communic

Bit 2: MP	Description	
0	Multiprocessor function disabled	(In
1	Multiprocessor format selected	

Bits 1 and 0—Clock Select 1 and 0 (CKS1, CKS0): Select the internal clock source o chip baud rate generator. Four clock sources are available. Pϕ, Pϕ/4, Pϕ/16 and Pϕ/64 c according to the setting of the CKS1 and CKS0 bits. For further information on the cloc bit rate register settings, and baud rate, see section 14.2.9, Bit Rate Register (SCBRR).

Bit 1: CKS1	Bit 0: CKS0	Description	
0	0	Pϕ	(In
	1	Pϕ/4	
1	0	Pϕ/16	
	1	Pϕ/64	

Note: Pϕ: Peripheral clock

Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7—Transmit Interrupt Enable (TIE): Enables or disables the transmit-data-emp
(TXI) requested when the transmit data register empty bit (TDRE) in the serial status r
(SCSSR) is set to 1 due to transfer of serial transmit data from SCTDR to SCTSR.

Bit 7: TIE	Description
0	Transmit-data-empty interrupt request (TXI) is disabled* (
1	Transmit-data-empty interrupt request (TXI) is enabled

Note: * The TXI interrupt request can be cleared by reading TDRE after it has been s
clearing TDRE to 0, or by clearing TIE to 0.

Bit 6—Receive Interrupt Enable (RIE): Enables or disables the receive-data-full int
requested when the receive data register full bit (RDRF) in the serial status register (S
to 1 due to transfer of serial receive data from SCRSR to SCRDR. It also enables or di
receive-error interrupt (ERI) requests.

Bit 6: RIE	Description
0	Receive-data-full interrupt (RXI) and receive-error interrupt (ERI) req disabled* (
1	Receive-data-full interrupt (RXI) and receive-error interrupt (ERI) req enabled

Note: * RXI and ERI interrupt requests can be cleared by reading the RDRF flag or er
(FER, PER, or ORER) after it has been set to 1, then clearing the flag to 0, or
RIE to 0.

Select the transmit format in SCSMR before setting TE to 1.

Bit 4—Receive Enable (RE): Enables or disables the SCI serial receiver.

Bit 4: RE	Description
0	Receiver disabled[*1] (In
1	Receiver enabled[*2]

Notes: 1. Clearing RE to 0 does not affect the receive flags (RDRF, FER, PER, ORER flags retain their previous values.
2. Serial reception starts when a start bit is detected in asynchronous mode, or synchronous clock input is detected in synchronous mode. Select the receiv SCSMR before setting RE to 1.

Bit 3—Multiprocessor Interrupt Enable (MPIE): Enables or disables multiprocessor The MPIE setting is used only in asynchronous mode, and only if the multiprocessor m (MP) in the serial mode register (SCSMR) is set to 1 during reception. The MPIE settin ignored in synchronous mode or when the MP bit is cleared to 0.

Bit 3: MPIE	Description
0	Multiprocessor interrupts are disabled (normal receive operation) (In [Clearing conditions] (1) MPE is cleared to 0 when MPIE is cleared to 0. (2) The multiprocessor bit (MPB) is set to 1 in receive data.
1	Multiprocessor interrupts are enabled* Receive-data-full interrupt requests (RXI), receive-error interrupt reque and setting of the RDRF, FER, and ORER status flags in the serial sta (SCSSR) are disabled until data with a multiprocessor bit of 1 is receiv

Note: * The SCI does not transfer receive data from SCRSR to SCRDR, does not dete errors, and does not set the RDRF, FER, and ORER flags in the serial status re (SCSSR). When it receives data that includes MPB = 1, the SCSSR's MPB flag and the SCI automatically clears MPIE to 0, generates RXI and ERI interrupts (and RIE bits in the SCSCR are set to 1), and allows the FER and ORER bits to

Bits 1 and 0—Clock Enable 1 and 0 (CKE1, CKE0): Select the SCI clock source an disable clock output from the SCK pin. Depending on the combination of CKE1 and C SCK pin can be used for serial clock output or serial clock input.

The CKE0 setting is valid only in asynchronous mode, and only when the SCI is intern clocked (CKE1 = 0). The CKE0 setting is ignored in synchronous mode, or when an e clock source is selected (CKE1 = 1). Before selecting the SCI operating mode in the s register (SCSMR), set CKE1 and CKE0. For further details on selection of the SCI clo see table 14.10 in section 14.3, Operation.

Bit 1: CKE1	Bit 0: CKE0	Description	
0	0	Asynchronous mode	Internal clock, SCK pin used for input pin is ignored) (In
		Synchronous mode	Internal clock, SCK pin used for synchron output (In
	1	Asynchronous mode	Internal clock, SCK pin used for clock out
		Synchronous mode	Internal clock, SCK pin used for synchron output
1	0	Asynchronous mode	External clock, SCK pin used for clock inp
		Synchronous mode	External clock, SCK pin used for synchro input
	1	Asynchronous mode	External clock, SCK pin used for clock inp
		Synchronous mode	External clock, SCK pin used for synchro input

Notes: 1. The output clock frequency is the same as the bit rate.
2. The input clock frequency is 16 times the bit rate.

Bit:	7	6	5	4	3	2	1
	TDRE	RDRF	ORER	FER	PER	TEND	MPB
Initial value:	1	0	0	0	0	1	0
R/W:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R	R

Note: * The only value that can be written is 0 to clear the flag.

Bit 7—Transmit Data Register Empty (TDRE): Indicates that the SCI has loaded tra
from SCTDR into SCTSR and new serial transmit data can be written in SCTDR.

Bit 7: TDRE	Description
0	SCTDR contains valid transmit data [Clearing condition] TDRE is cleared to 0 when software reads TDRE after it has been se
1	SCTDR does not contain valid transmit data (Ir [Setting conditions] (1) TDRE is set to 1 when the chip is reset or enters standby mode. (2) The TE bit in the serial control register (SCSCR) is cleared to 0. (3) SCTDR contents are loaded into SCTSR, so new data can be writ SCTDR.

	[Setting condition] RDRF is set to 1 when serial data is received normally and transferr SCRSR to SCRDR.

Note: SCRDR and RDRF are not affected by detection of receive errors or by clearing bit to 0 in the serial control register. They retain their previous contents. If RDR to 1 when reception of the next data ends, an overrun error (ORER) occurs and data is lost.

Bit 5—Overrun Error (ORER): Indicates that data reception aborted due to an over

Bit 5: ORER	Description
0	Receiving is in progress or has ended normally[*1] ([Clearing conditions] (1) ORER is cleared to 0 when the chip is reset or enters standby m (2) When software reads ORER after it has been set to 1, then write ORER.
1	A receive overrun error occurred[*2] [Setting condition] ORER is set to 1 if reception of the next serial data ends when RDR

Notes: 1. Clearing the RE bit to 0 in the serial control register does not affect the OR retains its previous value.

2. SCRDR continues to hold the data received before the overrun error, so su receive data is lost. Serial receiving cannot continue while ORER is set to 1 synchronous mode, serial transmitting is also disabled.

1	A receive framing error occurred [Setting condition] FER is set to 1 if the stop bit at the end of receive data is checked and be 0.*2

Notes: 1. Clearing the RE bit to 0 in the serial control register does not affect the FER retains its previous value.
2. When the stop bit length is two bits, only the first bit is checked. The second not checked. When a framing error occurs, the SCI transfers the receive data SCRDR but does not set RDRF. Serial receiving cannot continue while FER In synchronous mode, serial transmitting is also disabled.

Bit 3—Parity Error (PER): Indicates that data reception (with parity) aborted due to a error in asynchronous mode.

Bit 3: PER	Description
0	Receiving is in progress or has ended normally*1 (In [Clearing conditions] (1) PER is cleared to 0 when the chip is reset or enters standby mode (2) When software reads PER after it has been set to 1, then writes 0
1	A receive parity error occurred*2 [Setting condition] PER is set to 1 if the number of 1s in receive data, including the parity not match the even or odd parity setting of the parity mode bit (O/$\overline{E}$) in mode register (SCSMR).

Notes: 1. Clearing the RE bit to 0 in the serial control register does not affect the PER retains its previous value.
2. When a parity error occurs, the SCI transfers the receive data into SCRDR b set RDRF. Serial receiving cannot continue while PER is set to 1. In synchro mode, serial transmitting is also disabled.

1	End of transmission [Setting conditions] (1) TEND is set to 1 when the chip is reset or enters standby mode. (2) When TE is cleared to 0 in the serial control register (SCSCR). (3) If TDRE is 1 when the last bit of a one-byte serial character is tra	(

Bit 1—Multiprocessor Bit (MPB): Stores the value of the multiprocessor bit in recei
when a multiprocessor format is selected for receiving in asynchronous mode. MPB is
bit and cannot be written to.

Bit 1: MPB	Description	
0	Multiprocessor bit value in receive data is 0*	(
1	Multiprocessor bit value in receive data is 1	

Note: * If RE is cleared to 0 when a multiprocessor format is selected, MPB retains its
previous value.

Bit 0—Multiprocessor Bit Transfer (MPBT): Stores the value of the multiprocesso
transmit data when a multiprocessor format is selected for transmitting in asynchronou
The MPBT setting is ignored in synchronous mode, when a multiprocessor format is n
or when the SCI is not transmitting.

Bit 0: MPBT	Description	
0	Multiprocessor bit value in transmit data is 0	(
1	Multiprocessor bit value in transmit data is 1	

It is also possible to read data on the SCK pin, and write output data.

SCPCR

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2
	SCP7MD1	SCP7MD0	SCP6MD1	SCP6MD0	SCP5MD1	SCP5MD0	SCP4MD1	SCP4MD0	SCP3MD1	SCP3MD0	SCP2MD1	SCP2MD0	SCP1MD1	SCI MD
Initial value:	1	0	1	0	1	0	0	0	1	0	0	0	1	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/

SCPDR

Bit:	7	6	5	4	3	2	1
	SCP7DT	SCP6DT	SCP5DT	SCP4DT	SCP3DT	SCP2DT	SCP1D
Initial value:	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W

SCI pin I/O and data control are performed by bits 3–0 of SCPCR and bits 1 and 0 of S

SCPCR Bits 3 and 2—Serial Clock Port I/O (SCP1MD1, SCP1MD0): Specify seria pin I/O. When the SCK pin is actually used as a port I/O pin, clear the C/$\overline{A}$ bit in SCSM CKE1 and CKE0 in SCSCR to 0.

Bit 3: SCP1MD1	Bit 2: SCP1MD0	Description
0	0	SCP1DT bit value is not output to SCK pin
0	1	SCP1DT bit value is output to SCK pin
1	0	SCK pin value is read from SCP1DT bit
1	1	(Initial value

SCPCR Bits 1 and 0—Serial Port Break I/O (SCP0MD1, SCP0MD0): Specify the TxD pin output condition. When the TxD pin is actually used as a port output pin and value set with the SCP0DT bit, clear the TE bit in SCSCR to 0.

Bit 1: SCP0MD1	Bit 0: SCP0MD0	Description
0	0	SCP0DT bit value is not output to TxD pin
0	1	SCP0DT bit value is output to TxD pin

SCPDR Bit 0—Serial Port Break Data (SCP0DT): Specifies the serial port RxD pi and TxD pin output data. The TxD pin output condition is specified by the SCP0MD1 SCP0MD0 bits. When the TxD pin is set to output mode, the value of the SCP0DT bi the TxD pin. The RxD pin value is read from the SCP0DT bit regardless of the values SCP0MD1 and SCP0MD0 bits, if RE in SCSCR is set to 1. The initial value of this bi power-on reset is undefined.

Bit 0: SCP0DT	Description
0	I/O data is low
1	I/O data is high

Block diagrams of the SCI I/O port pins are shown in figures 14.2, 14.3, and 14.4.

Bit:	7	6	5	4	3	2	1
Initial value:	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The SCBRR setting is calculated as follows:

$$\text{Asynchronous mode: } N = \frac{P\phi}{64 \times 2^{2n-1} \times B} \times 10^6 - 1$$

$$\text{Synchronous mode: } N = \frac{P\phi}{8 \times 2^{2n-1} \times B} \times 10^6 - 1$$

B: Bit rate (bits/s)
N: SCBRR setting for baud rate generator ($0 \leq N \leq 255$)
Pϕ: Operating frequency for peripheral modules (MHz)
n: Baud rate generator clock source (n = 0, 1, 2, 3) (for the clock sources and n, see table 14.3.)

Table 14.3 SCSMR Settings

		SCSMR Settings	
n	**Clock Source**	**CKS1**	**CKS0**
0	Pϕ	0	0
1	Pϕ/4	0	1
2	Pϕ/16	1	0
3	Pϕ/64	1	1

Note: The bit rate error in asynchronous is given by the following formula:

$$\text{Error } (\%) = \left(\frac{P\phi \times 10^6}{(N+1) \times B \times 64 \times 2^{2n-1}}\right) \times 100$$

150	1	103	0.16	1	108	0.21	1	127
300	0	207	0.16	0	217	0.21	0	255
600	0	103	0.16	0	108	0.21	0	127
1200	0	51	0.16	0	54	–0.70	0	63
2400	0	25	0.16	0	26	1.14	0	31
4800	0	12	0.16	0	13	–2.48	0	15
9600	0	6	–6.99	0	6	–2.48	0	7
19200	0	2	8.51	0	2	13.78	0	3
31250	0	1	0.00	0	1	4.86	0	1
38400	0	1	–18.62	0	1	–14.67	0	1

	Pϕ (MHz)							
	3			3.6864			4	
Bit Rate (bits/s)	**n**	**N**	**Error (%)**	**n**	**N**	**Error (%)**	**n**	**N**
110	1	212	0.03	2	64	0.70	2	70
150	1	155	0.16	1	191	0.00	1	207
300	1	77	0.16	1	95	0.00	1	103
600	0	155	0.16	0	191	0.00	0	207
1200	0	77	0.16	0	95	0.00	0	103
2400	0	38	0.16	0	47	0.00	0	51
4800	0	19	–2.34	0	23	0.00	0	25
9600	0	9	–2.34	0	11	0.00	0	12
19200	0	4	–2.34	0	5	0.00	0	6
31250	0	2	0.00	—	—	—	0	3
38400	—	—	—	0	2	0.00	0	2

2400	0	63	0.00	0	64	0.16	0	77
4800	0	31	0.00	0	32	–1.36	0	38
9600	0	15	0.00	0	15	1.73	0	19
19200	0	7	0.00	0	7	1.73	0	9
31250	0	4	–1.70	0	4	0.00	0	5
38400	0	3	0.00	0	3	1.73	0	4

	Pϕ (MHz)							
	6.144			7.3728			8	
Bit Rate (bits/s)	**n**	**N**	**Error (%)**	**n**	**N**	**Error (%)**	**n**	**N**
110	2	108	0.08	2	130	–0.07	2	141
150	2	79	0.00	2	95	0.00	2	103
300	1	159	0.00	1	191	0.00	1	207
600	1	79	0.00	1	95	0.00	1	103
1200	0	159	0.00	0	191	0.00	0	207
2400	0	79	0.00	0	95	0.00	0	103
4800	0	39	0.00	0	47	0.00	0	51
9600	0	19	0.00	0	23	0.00	0	25
19200	0	9	0.00	0	11	0.00	0	12
31250	0	5	2.40	0	6	5.33	0	7
38400	0	4	0.00	0	5	0.00	0	6

1200	1	95	0.00	1	103	0.16	1	127	0.00	1	1
2400	0	191	0.00	0	207	0.16	0	255	0.00	1	6
4800	0	95	0.00	0	103	0.16	0	127	0.00	0	1
9600	0	47	0.00	0	51	0.16	0	63	0.00	0	6
19200	0	23	0.00	0	25	0.16	0	31	0.00	0	3
31250	0	14	–1.70	0	15	0.00	0	19	–1.70	0	1
38400	0	11	0.00	0	12	0.16	0	15	0.00	0	1

	Pϕ (MHz)										
	24			24.576			28.7				
Bit Rate (bits/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N
110	3	106	–0.44	3	108	0.08	3	126	0.31	3	1
150	3	77	0.16	3	79	0.00	3	92	0.46	3	9
300	2	155	0.16	2	159	0.00	2	186	–0.08	2	1
600	2	77	0.16	2	79	0.00	2	92	0.46	2	9
1200	1	155	0.16	1	159	0.00	1	186	–0.08	1	1
2400	1	77	0.16	1	79	0.00	1	92	0.46	1	9
4800	0	155	0.16	0	159	0.00	0	186	–0.08	0	1
9600	0	77	0.16	0	79	0.00	0	92	0.46	0	9
19200	0	38	0.16	0	39	0.00	0	46	–0.61	0	4
31250	0	23	0.00	0	24	–1.70	0	28	–1.03	0	2
38400	0	19	–2.34	0	19	0.00	0	22	1.55	0	2

1k	1	249	2	124	2	249	3	111	3
2.5k	1	99	1	199	2	99	2	178	2
5k	0	199	1	99	1	199	2	89	2
10k	0	99	0	199	1	99	1	178	1
25k	0	39	0	79	0	159	1	71	1
50k	0	19	0	39	0	79	0	143	0
100k	0	9	0	19	0	39	0	71	0
250k	0	3	0	7	0	15	—	—	0
500k	0	1	0	3	0	7	—	—	0
1M	0	0*	0	1	0	3	—	—	—
2M			0	0*	0	1	—	—	—

Notes: Settings with an error of 1% or less are recommended.

Blank: No setting possible

—: Setting possible, but error occurs

*: Continuous transmit/receive operation not possible

2.097152	65536	0	0
2.4576	76800	0	0
3	93750	0	0
3.6864	115200	0	0
4	125000	0	0
4.9152	153600	0	0
8	250000	0	0
9.8304	307200	0	0
12	375000	0	0
14.7456	460800	0	0
16	500000	0	0
19.6608	614400	0	0
20	625000	0	0
24	750000	0	0
24.576	768000	0	0
28.7	896875	0	0
30	937500	0	0

4	1.0000	62500
4.9152	1.2288	76800
8	2.0000	125000
9.8304	2.4576	153600
12	3.0000	187500
14.7456	3.6864	230400
16	4.0000	250000
19.6608	4.9152	307200
20	5.0000	312500
24	6.0000	375000
24.576	6.1440	384000
28.7	7.1750	448436
30	7.5000	468750

Table 14.8 Maximum Bit Rates with External Clock Input (Synchronous Mode)

Pϕ (MHz)	External Input Clock (MHz)	Maximum Bit Rate (
8	1.3333	1333333.3
16	2.6667	2666666.7
24	4.0000	4000000.0
28.7	4.7833	4783333.3
30	5.0000	5000000.0

the serial control register (SCSCR), as shown in table 14.10.

Asynchronous Mode:

- Data length is selectable: 7 or 8 bits.
- Parity and multiprocessor bits are selectable. So is the stop bit length (1 or 2 bits). combination of the preceding selections constitutes the communication format and length.
- In receiving, it is possible to detect framing errors (FER), parity errors (PER), over (ORER) and breaks.
- An internal or external clock can be selected as the SCI clock source.
  - When an internal clock is selected, the SCI operates using the on-chip baud rat and can output a serial clock signal with a frequency matching the bit rate.
  - When an external clock is selected, the external clock input must have a freque the bit rate. (The on-chip baud rate generator is not used.)

Synchronous Mode:

- The transmission/reception format has a fixed 8-bit data length.
- In receiving, it is possible to detect overrun errors (ORER).
- An internal or external clock can be selected as the SCI clock source.
  - When an internal clock is selected, the SCI operates using the on-chip baud rat and outputs a serial clock signal to external devices.
  - When an external clock is selected, the SCI operates on the input serial clock. T baud rate generator is not used.

	1	0		0		7-bit	Not set	
				1				
		1		0			Set	
				1				
	0	*	1	0	Asynchronous (multiprocessor format)	8-bit	Not set	Set
		*		1				
	1	*		0		7-bit		
		*		1				
1	*	*	*	*	Synchronous	8-bit		Not set

Note: Asterisks (*) indicate don't care bits.

Table 14.10 SCSMR and SCSCR Settings and SCI Clock Source Selection

SCSMR	SCSCR Settings			SCI Transmit/Receive Clock	
Bit 7 C/$\overline{A}$	Bit 1 CKE1	Bit 0 CKE0	Mode	Clock Source	SCK Pin Function
0	0	0	Asynchronous mode	Internal	SCI does not use the S
		1			Outputs a clock with fre matching the bit rate
	1	0		External	Inputs a clock with freq times the bit rate
		1			
1	0	0	Synchronous mode	Internal	Outputs the synchronou
		1			
	1	0		External	Inputs the synchronous
		1			

Figure 14.5 shows the general format of asynchronous serial communication. In asynchr
serial communication, the communication line is normally held in the mark (high) stat
monitors the line and starts serial communication when the line goes to the space (low
indicating a start bit. One serial character consists of a start bit (low), data (LSB first; s
the lowerest bit), parity bit (high or low), and stop bit (high), in that order.

When receiving in asynchronous mode, the SCI synchronizes at the falling edge of the
The SCI samples each data bit on the eighth pulse of a clock with a frequency 16 time
Receive data is latched at the center of each bit.

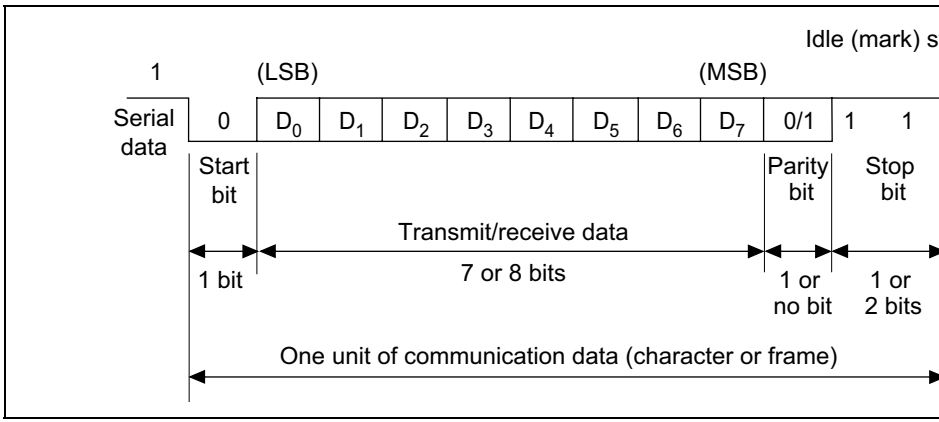


**Figure 14.5 Example of Data Format in Asynchronous Communicatio
(8-Bit Data with Parity and Two Stop Bits)**

0	0	0	1	START	8-bit data		STOP	ST
0	1	0	0	START	8-bit data		P	ST
0	1	0	1	START	8-bit data		P	ST
1	0	0	0	START	7-bit data	STOP		
1	0	0	1	START	7-bit data	STOP	STOP	
1	1	0	0	START	7-bit data	P	STOP	
1	1	0	1	START	7-bit data	P	STOP	ST
0	—	1	0	START	8-bit data		MPB	ST
0	—	1	1	START	8-bit data		MPB	ST
1	—	1	0	START	7-bit data	MPB	STOP	
1	—	1	1	START	7-bit data	MPB	STOP	ST

Notes: — : Don't care bits
START: Start bit
STOP: Stop bit
P: Parity bit
MPB: Multiprocessor bit

Clock: An internal clock generated by the on-chip baud rate generator or an external cl
from the SCK pin can be selected as the SCI transmit/receive clock. The clock source is
by the C/$\overline{A}$ bit in the serial mode register (SCSMR) and bits CKE1 and CKE0 in the ser
register (SCSCR) (table 14.10).

When an external clock is input at the SCK pin, it must have a frequency equal to 16 tir
desired bit rate.

Figure 14.6 Output Clock and Serial Data Timing (Asynchronous Mo

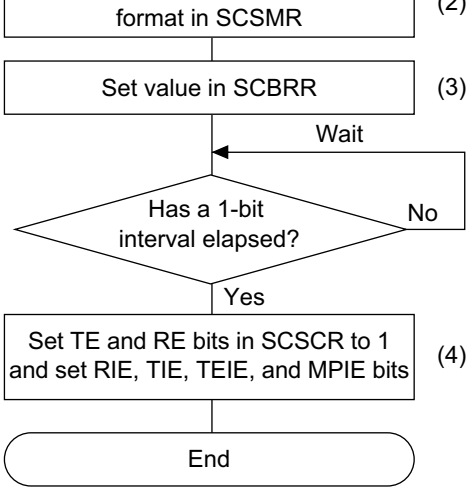
Transmitting and Receiving Data (SCI Initialization (Asynchronous Mode)): Bef
transmitting or receiving, clear the TE and RE bits to 0 in the serial control register (S
initialize the SCI as follows.

When changing the operation mode or communication format, always clear the TE an
0 before following the procedure given below. Clearing TE to 0 sets TDRE to 1 and in
transmit shift register (SCTSR). Clearing RE to 0, however, does not initialize the RD
FER, and ORER flags or receive data register (SCRDR), which retain their previous c

When an external clock is used, the clock should not be stopped during initialization o
operation. SCI operation becomes unreliable if the clock is stopped.

Figure 14.7 shows a sample flowchart for initializing the SCI. The procedure for initia
SCI is:

1. Select the clock source in the serial control register (SCSCR). Leave RIE, TIE, TE
TE, and RE cleared to 0. If clock output is selected in asynchronous mode, clock o
immediately after the setting is made in SCSCR.
2. Select the communication format in the serial mode register (SCSMR).
3. Write the value corresponding to the bit rate in the bit rate register (SCBRR) (not r
an external clock is used).
4. Wait for at least the interval required to transmit or receive one bit, then set TE or
serial control register (SCSCR) to 1. Also set RIE, TIE, TEIE, and MPIE as neces
TE or RE enables the SCI to use the TxD or RxD pin. The initial state is the mark
transmitting, or the idle state (waiting for a start bit) when receiving.

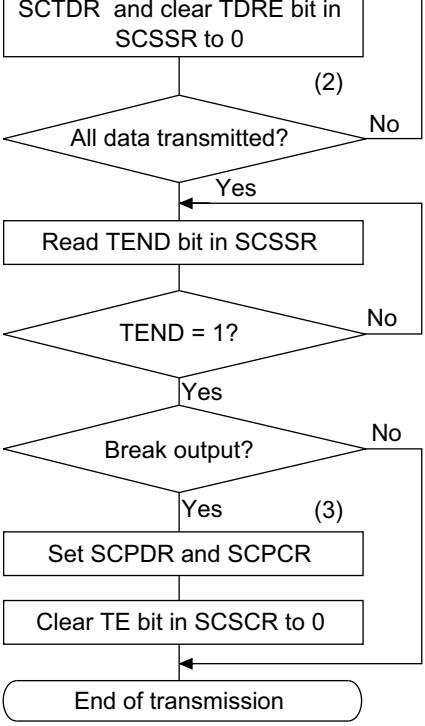


Note: Numbers in parentheses refer to steps in the preceding procedure description.

Figure 14.7 Sample Flowchart for SCI Initialization

Transmitting Serial Data (Asynchronous Mode): Figure 14.8 shows a sample flowcl transmitting serial data. The procedure for transmitting serial data is:

1. SCI status check and transmit data write: Read the serial status register (SCSSR), ch the TDRE bit is 1, then write transmit data in the transmit data register (SCTDR) an TDRE to 0.
2. To continue transmitting serial data: Read the TDRE bit to check whether it is safe it reads 1); if so, write data in SCTDR, then clear TDRE to 0.
3. To output a break at the end of serial transmission: Set the port SC data register (SC port SC control register (SCPCR), then clear the TE bit to 0 in the serial control reg (SCSCR). For SCPCR and SCPDR settings, see section 14.2.8, SC Port Control Re (SCPCR)/SC Port Data Register (SCPDR).



Note: Numbers in parentheses refer to steps in the preceding procedure description.

Figure 14.8 Sample Flowchart for Transmitting Serial Data

a. Start bit: One 0 bit is output.
b. Transmit data: Seven or eight bits of data are output, LSB first.
c. Parity bit or multiprocessor bit: One parity bit (even or odd parity) or one multip bit is output. Formats in which neither a parity bit nor a multiprocessor bit is out also be selected.
d. Stop bit: One or two 1-bits (stop bits) are output.
e. Marking: Output of 1-bits continues until the start bit of the next transmit data.

3. The SCI checks the TDRE bit when it outputs the stop bit. If TDRE is 0, the SCI loa data from SCTDR into SCTSR, outputs the stop bit, then begins serial transmission frame. If TDRE is 1, the SCI sets the TEND bit to 1 in SCSSR, outputs the stop bit, continues output of 1-bits (marking). If the transmit-end interrupt enable bit (TEIE) is set to 1, a transmit-end interrupt (TEI) is requested.

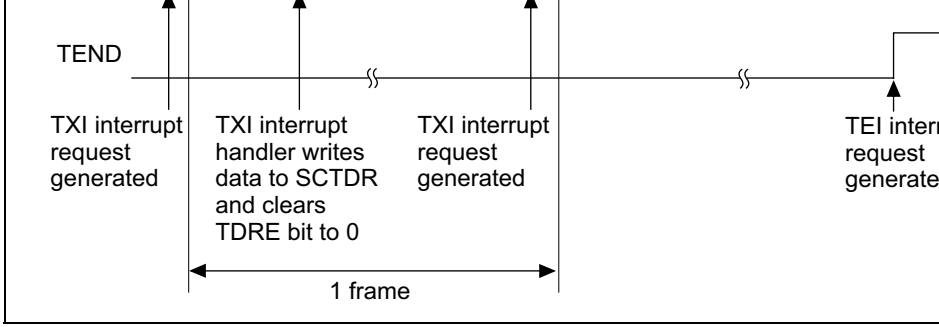
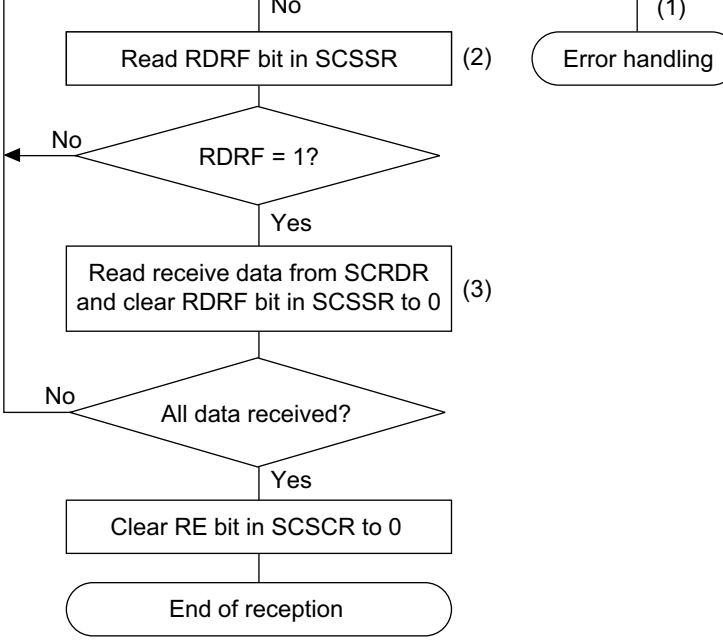


Figure 14.9 Example of SCI Transmit Operation in Asynchronous Mo (8-Bit Data with Parity and One Stop Bit)

Receiving Serial Data (Asynchronous Mode): Figure 14.10 shows a sample flowcha receiving serial data. The procedure for receiving serial data after enabling the SCI for is:

1. Receive error handling and break detection: If a receive error occurs, read the ORE FER bits in SCSSR to identify the error. After executing the necessary error handli ORER, PER and FER to 0. Receiving cannot resume if ORER, PER or FER remai When a framing error occurs, the RxD pin can be read to detect the break state.
2. SCI status check and receive-data read: Read the serial status register (SCSSR), ch RDRF is set to 1, then read receive data from the receive data register (SCRDR) a RDRF to 0. The RXI interrupt can also be used to determine if the RDRF bit has c 0 to 1.
3. To continue receiving serial data: Read the RDRF and SCRDR bits and clear RDR before the stop bit of the current frame is received.



Note: Numbers in parentheses refer to steps in the preceding procedure description.

Figure 14.10 Sample Flowchart for Receiving Serial Data

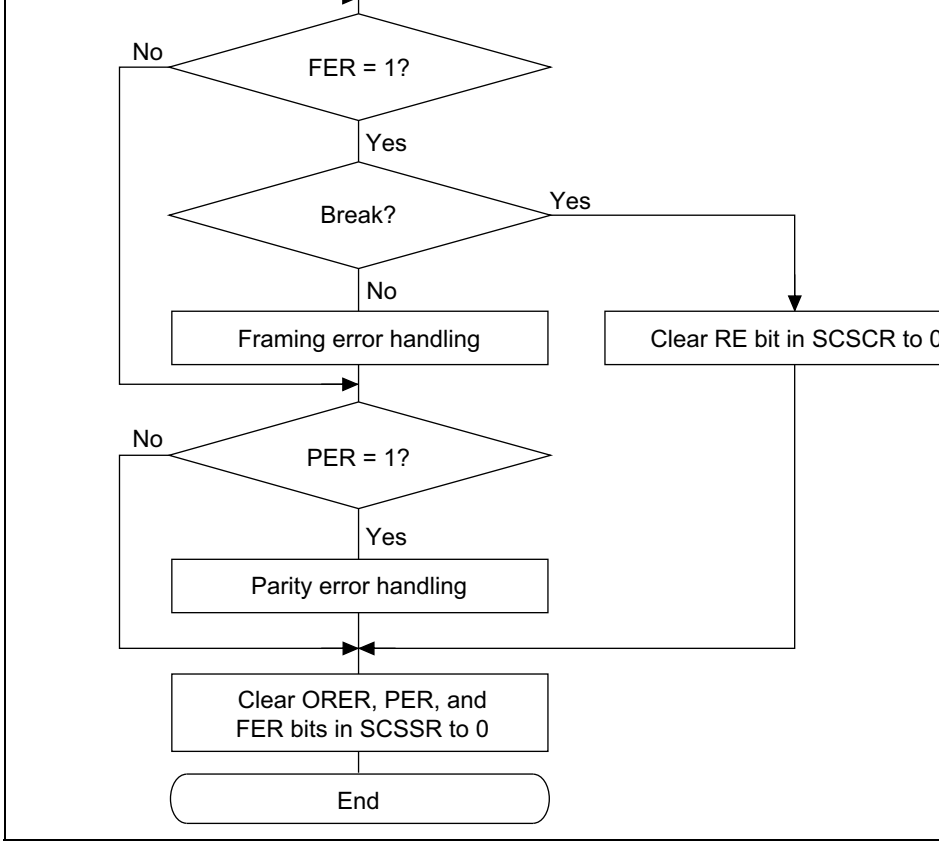


Figure 14.10 Sample Flowchart for Receiving Serial Data (cont)

b. Stop bit check: The stop bit value must be 1. If there are two stop bits, only the f
is checked.

c. Status check: RDRF must be 0 so that receive data can be loaded from SCRSR i
SCRDR.

If these checks all pass, the SCI sets RDRF to 1 and stores the received data in S
one of the checks fails (receive error), the SCI operates as indicated in table 14.1

Note: When a receive error flag is set, further receiving is disabled. The RDRF bit is r
Be sure to clear the error flags.

4. After setting RDRF to 1, if the receive-data-full interrupt enable bit (RIE) is set to 1
SCSCR, the SCI requests a receive-data-full interrupt (RXI). If one of the error flag
PER, or FER) is set to 1 and the receive-data-full interrupt enable bit (RIE) in SCSC
set to 1, the SCI requests a receive-error interrupt (ERI).

Table 14.12 Receive Error Conditions and SCI Operation

Receive Error	Abbreviation	Condition	Data Transfer
Overrun error	ORER	Receiving of next data ends while RDRF is still set to 1 in SCSSR	Receive data not transferred from into SCRDR
Framing error	FER	Stop bit is 0	Receive data tra from SCRSR into
Parity error	PER	Parity of receive data differs from even/odd parity setting in SCSMR	Receive data tra from SCRSR into

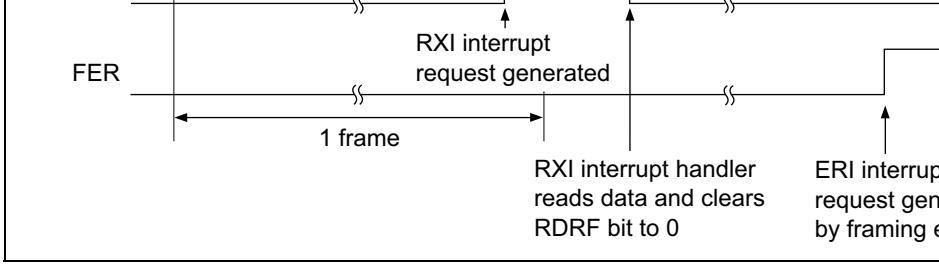


Figure 14.11 Example of SCI Receive Operation (8-Bit Data with Parity and One Stop Bit)

14.3.3 Multiprocessor Communication

The multiprocessor communication function enables several processors to share a sing communication line. The processors communicate in asynchronous mode using a form additional multiprocessor bit (multiprocessor format).

In multiprocessor communication, each receiving processor is addressed by a unique I communication cycle consists of an ID-sending cycle that identifies the receiving proc data-sending cycle. The multiprocessor bit distinguishes ID-sending cycles from data- cycles. The transmitting processor starts by sending the ID of the receiving processor it wants to communicate as data with the multiprocessor bit set to 1. Next the transmit processor sends transmit data with the multiprocessor bit cleared to 0.

Receiving processors skip incoming data until they receive data with the multiprocess 1. When they receive data with the multiprocessor bit set to 1, receiving processors co data with their IDs. The receiving processor with a matching ID continues to receive f incoming data. Processors with IDs not matching the received data skip further incom until they again receive data with the multiprocessor bit set to 1. Multiple processors c receive data in this way.

Figure 14.12 shows an example of communication among processors using the multip format.

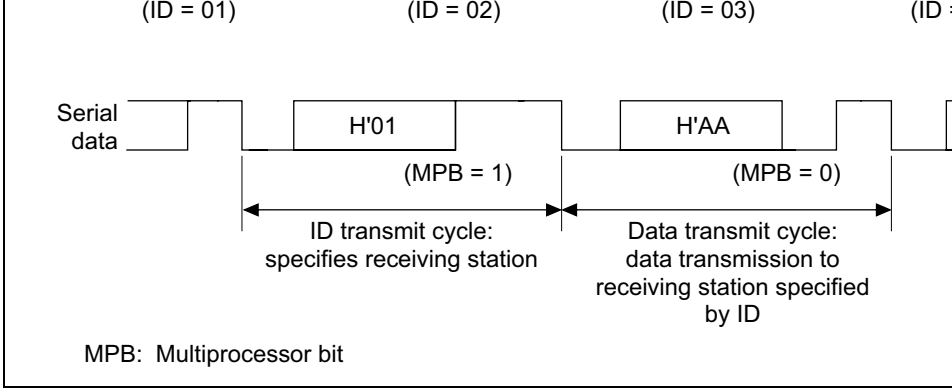


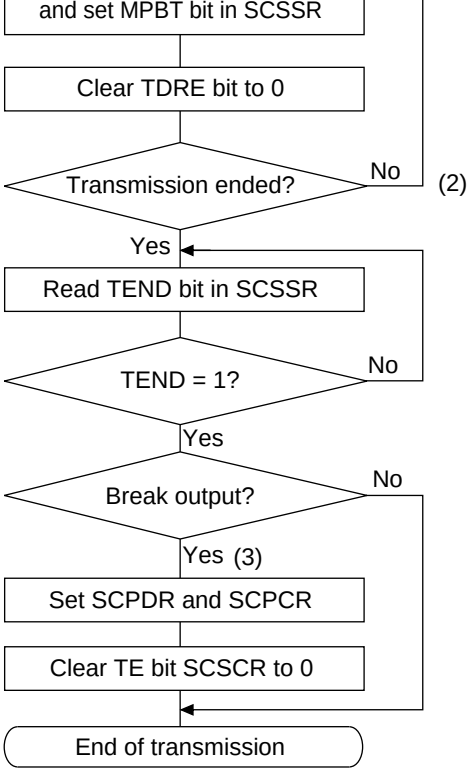
Figure 14.12 Communication Among Processors Using Multiprocessor For (Sending Data H'AA to Receiving Processor A)

Communication Formats: Four formats are available. Parity-bit settings are ignored w multiprocessor format is selected. For details see table 14.11.

Clock: See the description in the asynchronous mode section.

Transmitting Multiprocessor Serial Data: Figure 14.13 shows a sample flowchart fo transmitting multiprocessor serial data. The procedure for transmitting multiprocessor s is:

1. SCI status check and transmit data write: Read the serial status register (SCSSR), ch the TDRE bit is 1, then write transmit data in the transmit data register (SCTDR). A MPBT (multiprocessor bit transfer) to 0 or 1 in SCSSR. Finally, clear TDRE to 0.
2. To continue transmitting serial data: Read the TDRE bit to check whether it is safe t it reads 1); if so, write data in SCTDR, then clear TDRE to 0.
3. To output a break at the end of serial transmission: Set the port SC data register (SC port SC control register (SCPCR), then clear the TE bit to 0 in the serial control reg (SCSCR). For SCPCR and SCPDR settings, see section 14.2.8, SC Port Control Re (SCPCR)/SC Port Data Register (SCPDR).



Note: Numbers in parentheses refer to steps in the preceding procedure description.

Figure 14.13 Sample Flowchart for Transmitting Multiprocessor Serial

a. Start bit: One 0-bit is output.
b. Transmit data: Seven or eight bits are output, LSB first.
c. Multiprocessor bit: One multiprocessor bit (MPBT value) is output.
d. Stop bit: One or two 1-bits (stop bits) are output.
e. Marking: Output of 1-bits continues until the start bit of the next transmit data.

3. The SCI checks the TDRE bit when it outputs the stop bit. If TDRE is 0, the SCI tra from SCTDR into SCTSR, outputs the stop bit, then begins serial transmission of th frame. If TDRE is 1, the SCI sets the TEND bit in SCSSR to 1, outputs the stop bit, continues output of 1 bits in the mark state. If the transmit-end interrupt enable bit (SCSCR is set to 1, a transmit-end interrupt (TEI) is requested at this time.

Figure 14.14 shows SCI transmission with a multiprocessor format.

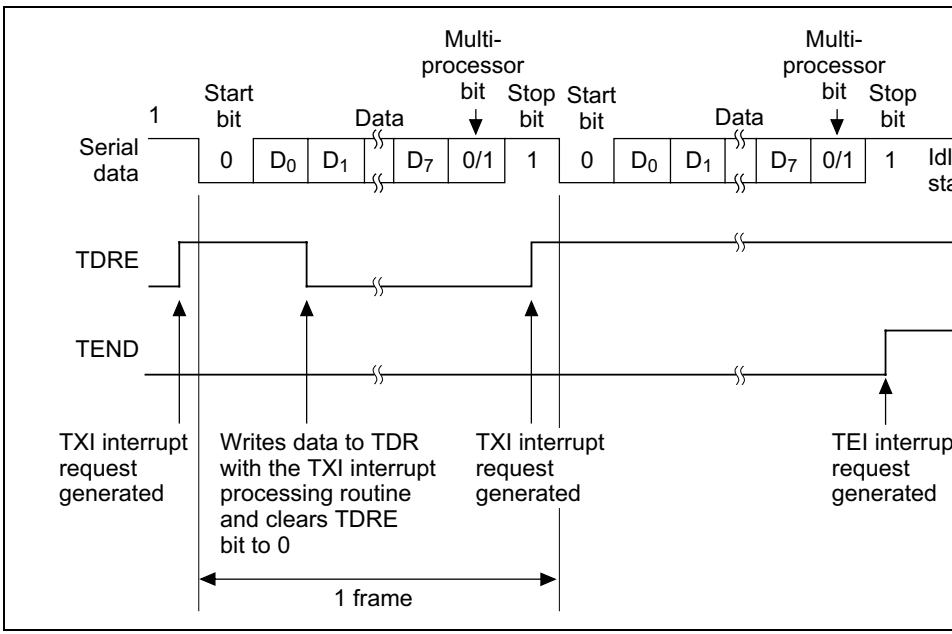
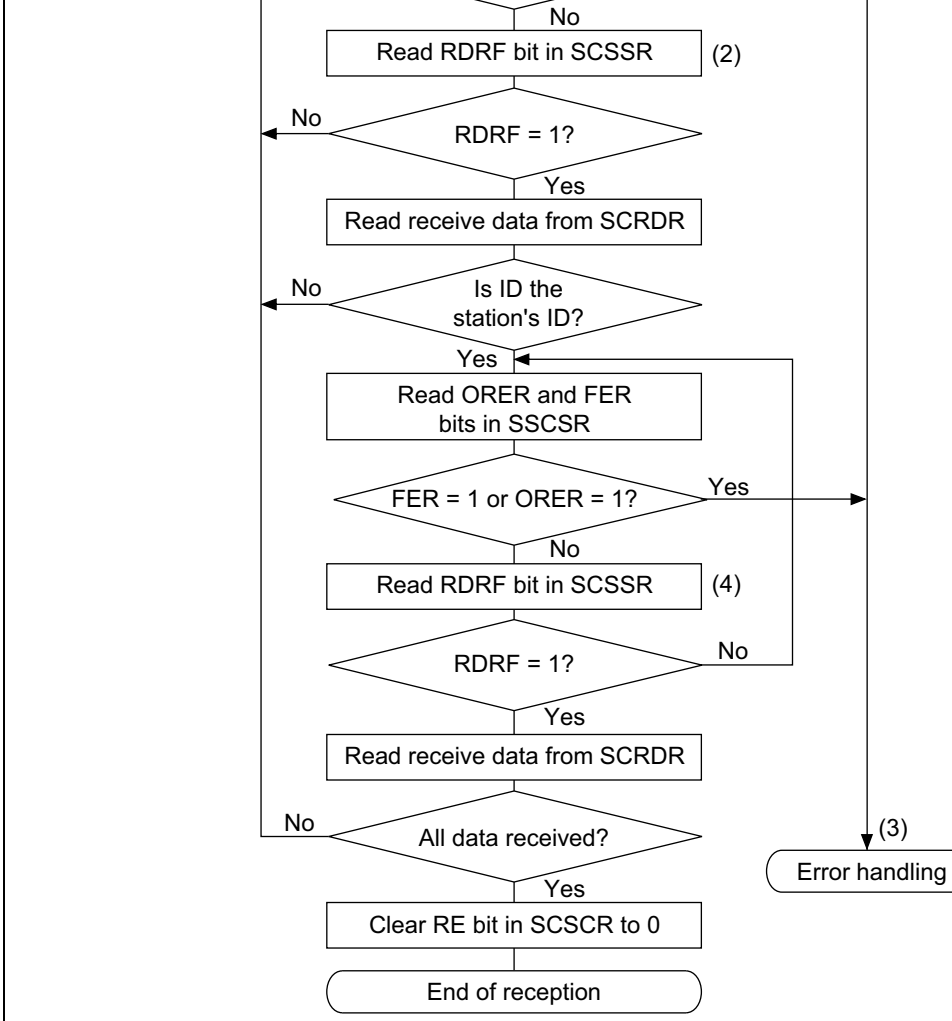


Figure 14.14 Example of SCI Multiprocessor Transmit Operation (8-Bit Data with Multiprocessor Bit and One Stop Bit)

from the receive data register (SCRDR).

4. Receive error handling and break detection: If a receive error occurs, read the ORI bits in SCSSR to identify the error. After executing the necessary error handling, c ORER and FER to 0. Receiving cannot resume if ORER or FER remain set to 1. W framing error occurs, the RxD pin can be read to detect the break state.



Note: Numbers in parentheses refer to steps in the preceding procedure description.

Figure 14.15 Sample Flowchart for Receiving Multiprocessor Serial Da

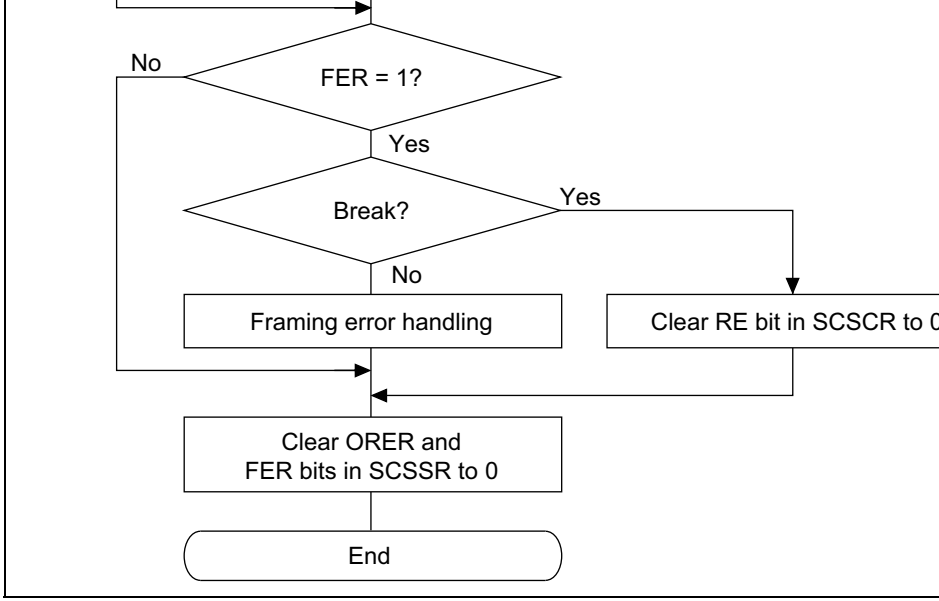


Figure 14.15 Sample Flowchart for Receiving Multiprocessor Serial Data

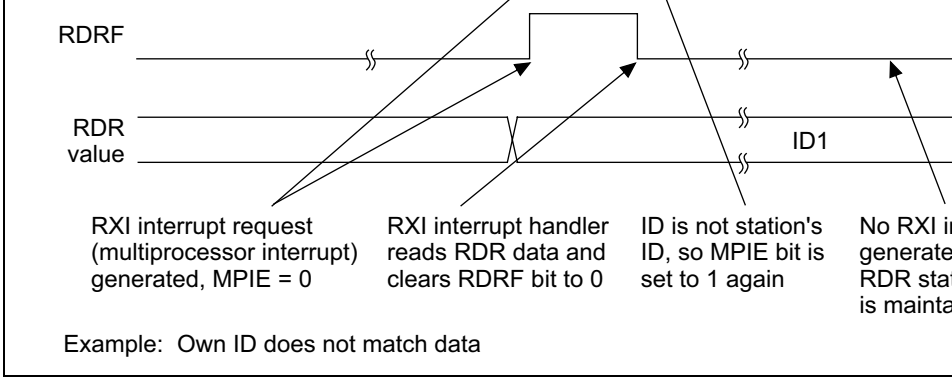


Figure 14.16 Example of SCI Receive Operation (8-Bit Data with Multiprocessor Bit and One Stop Bit)

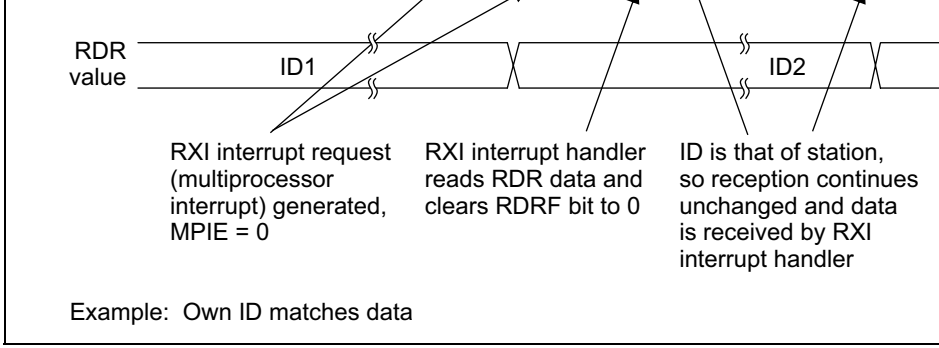


Figure 14.16 Example of SCI Receive Operation (cont)
(8-Bit Data with Multiprocessor Bit and One Stop Bit)

Figure 14.17 shows the general format in synchronous serial communication.

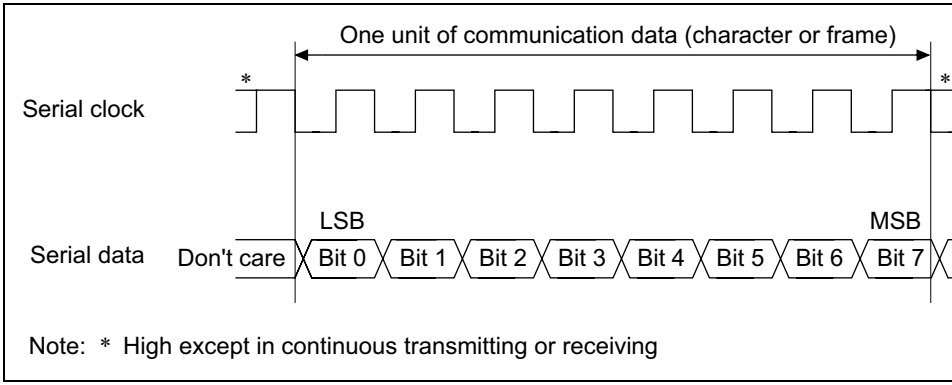


Figure 14.17 Data Format in Synchronous Communication

In synchronous serial communication, each data bit is output on the communication lin falling edge of the serial clock to the next. Data is guaranteed valid at the rising edge of clock. In each character, the serial data bits are transmitted in order from the LSB (first) MSB (last). After output of the MSB, the communication line remains in the state of th synchronous mode, the SCI transmits or receives data by synchronizing with the falling the serial clock.

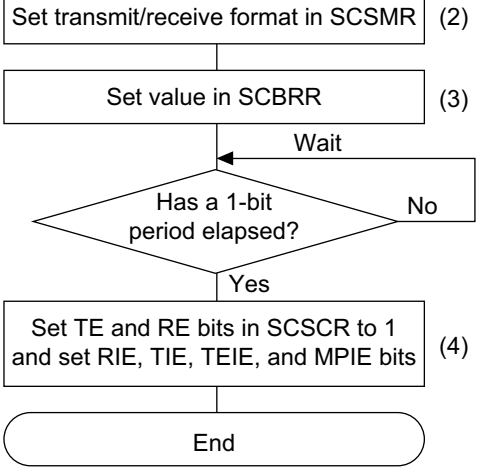
Communication Format: The data length is fixed at eight bits. No parity bit or multip can be added.

external clock source.

Transmitting and Receiving Data SCI Initialization (Synchronous Mode): Before transmitting, receiving, or changing the mode or communication format, the software the TE and RE bits to 0 in the serial control register (SCSCR), then initialize the SCI. to 0 sets TDRE to 1 and initializes the transmit shift register (SCTSR). Clearing RE to does not initialize the RDRF, PER, FER, and ORER flags and receive data register (S which retain their previous contents.

Figure 14.18 shows a sample flowchart for initializing the SCI. The procedure for init SCI is:

1. Select the clock source in the serial control register (SCSCR). Leave RIE, TIE, TE TE and RE cleared to 0.
2. Select transmit/receive format in the serial mode register (SCSMR).
3. Write the value corresponding to the bit rate in the bit rate register (SCBRR) (not r an external clock is used).
4. Wait for at least the interval required to transmit or receive one bit, then set TE or serial control register (SCSCR) to 1. Also set RIE, TIE, TEIE and MPIE. Setting T allows use of the TxD and RxD pins.

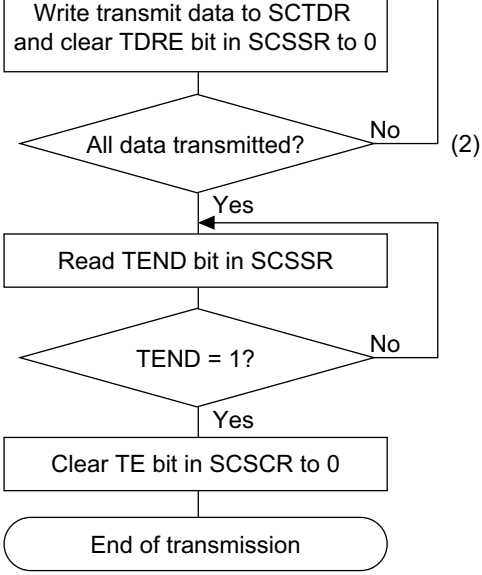


Note: Numbers in parentheses refer to steps in the preceding procedure description.

Figure 14.18 Sample Flowchart for SCI Initialization

Transmitting Serial Data (Synchronous Mode): Figure 14.19 shows a sample flowch transmitting serial data. The procedure for transmitting serial data is:

1. SCI status check and transmit data write: Read the serial status register (SCSSR), ch the TDRE bit is 1, then write transmit data in the transmit data register (SCTDR) an TDRE to 0.
2. To continue transmitting serial data: Read the TDRE bit to check whether it is safe it reads 1); if so, write data in SCTDR, then clear TDRE to 0.



Note: Numbers in parentheses refer to steps in the preceding procedure description.

Figure 14.19 Sample Flowchart for Transmitting Serial Data

clock source is selected, the SCI outputs data in synchronization with the input cloc
output from the TxD pin in order from the LSB (bit 0) to the MSB (bit 7).

3. The SCI checks the TDRE bit when it outputs the MSB (bit 7). If TDRE is 0, the SC
data from SCTDR into SCTSR, then begins serial transmission of the next frame. If
1, the SCI sets the TEND bit in SCSSR to 1, transmits the MSB, then holds the tran
pin (TxD) in the MSB state. If the transmit-end interrupt enable bit (TEIE) in SCSC
1, a transmit-end interrupt (TEI) is requested at this time.
4. After the end of serial transmission, the SCK pin is held in the high state.

Figure 14.20 shows an example of SCI transmit operation.

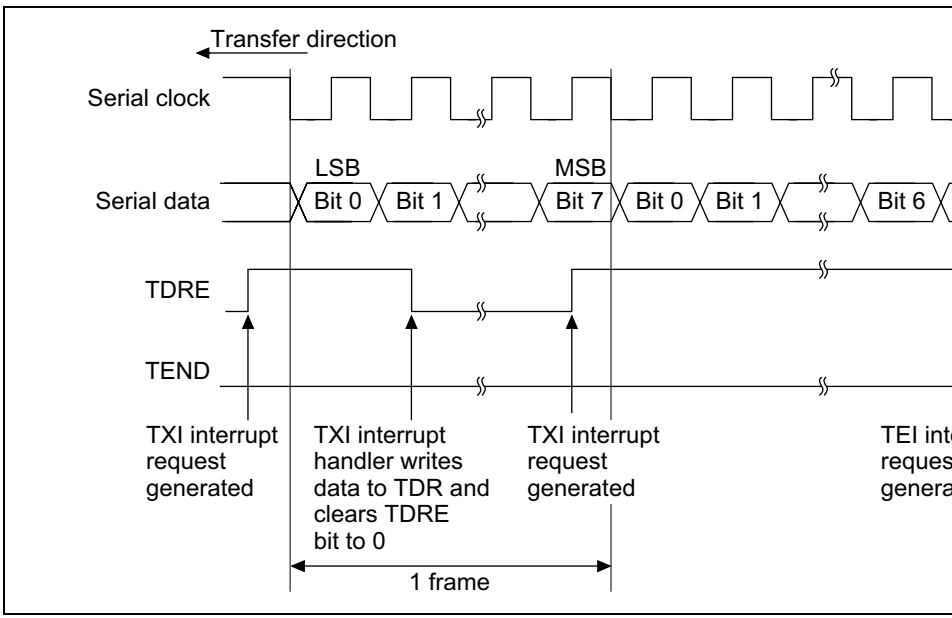
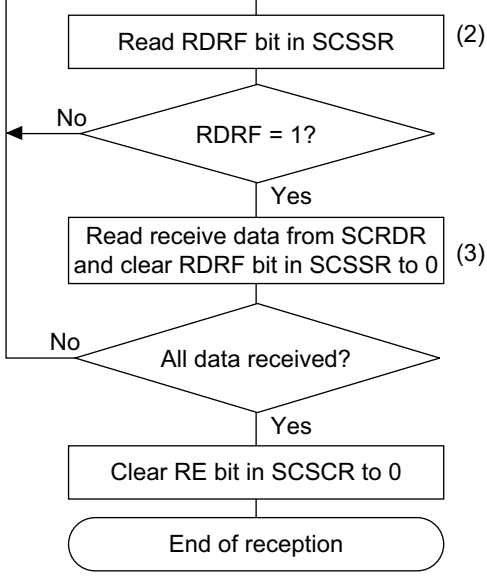


Figure 14.20 Example of SCI Transmit Operation

cannot resume if ORER remains set to 1.

2. SCI status check and receive data read: Read the serial status register (SCSSR), ch
   RDRF is set to 1, then read receive data from the receive data register (SCRDR) a
   RDRF to 0. The RXI interrupt can also be used to determine if the RDRF bit has c
   0 to 1.
3. To continue receiving serial data: Read SCRDR, and clear RDRF to 0 before the M
   of the current frame is received.



Note: Numbers in parentheses refer to steps in the preceding procedure description.

Figure 14.21 Sample Flowchart for Receiving Serial Data

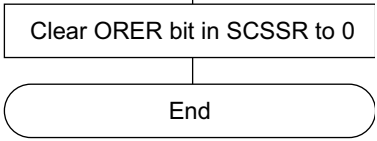


Figure 14.21 Sample Flowchart for Receiving Serial Data (cont)

In receiving, the SCI operates as follows:

1. The SCI synchronizes with serial clock input or output and initializes internally.
2. Receive data is shifted into SCRSR in order from the LSB to the MSB. After recei data, the SCI checks that RDRF is 0 so that receive data can be loaded from SCRS SCRDR. If this check is passed, the SCI sets RDRF to 1 and stores the received da SCRDR. If the check is not passed (receive error), the SCI operates as indicated in This state prevents further transmission or reception. While receiving, the RDRF b to 1. Be sure to clear the error flag.
3. After setting RDRF to 1, if the receive-data-full interrupt enable bit (RIE) is set to SCSCR, the SCI requests a receive-data-full interrupt (RXI). If the ORER bit is se receive-data-full interrupt enable bit (RIE) in SCSCR is also set to 1, the SCI requ receive-error interrupt (ERI).

Figure 14.22 shows an example of SCI receive operation.

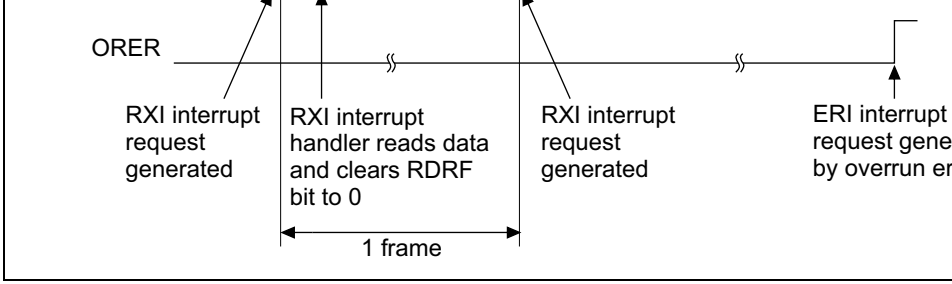
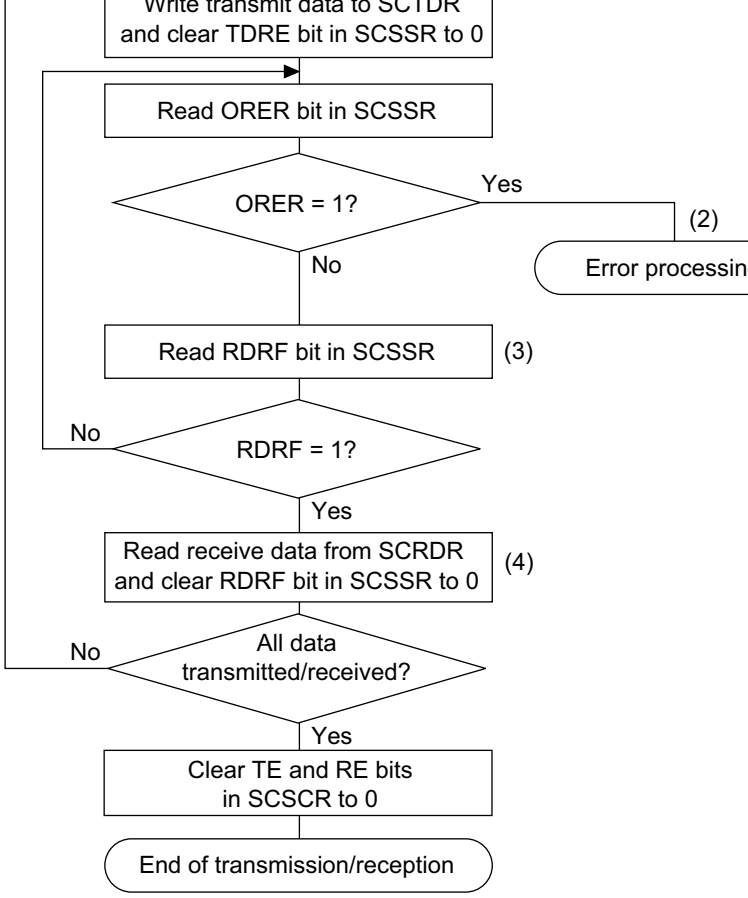


Figure 14.22 Example of SCI Receive Operation

Transmitting and Receiving Serial Data Simultaneously (Synchronous Mode): Fig shows a sample flowchart for transmitting and receiving serial data simultaneously. Th for setting the SCI to transmit and receive serial data simultaneously is:

1. SCI status check and transmit data write: Read the serial status register (SCSSR), ch the TDRE bit is 1, then write transmit data in the transmit data register (SCTDR) an TDRE to 0. The TXI interrupt can also be used to determine if the TDRE bit has ch 0 to 1.
2. Receive error handling: If a receive error occurs, read the ORER bit in SCSSR to id error. After executing the necessary error handling, clear ORER to 0. Transmitting/ cannot resume if ORER remains set to 1.
3. SCI status check and receive data read: Read the serial status register (SCSSR), che RDRF is set to 1, then read receive data from the receive data register (SCRDR) an RDRF to 0. The RXI interrupt can also be used to determine if the RDRF bit has ch 0 to 1.
4. To continue transmitting and receiving serial data: Read the RDRF bit and SCRDR, RDRF to 0 before the MSB (bit 7) of the current frame is received. Also read the TI check whether it is safe to write (if it reads 1); if so, write data in SCTDR, then clea 0 before the MSB (bit 7) of the current frame is transmitted.



Notes: 1. Numbers in parentheses refer to steps in the preceding procedure descripti
2. In switching from transmitting or receiving to simultaneous transmitting and receiving, clear both TE and RE to 0, then set both TE and RE to 1 sim

Figure 14.23 Sample Flowchart for Transmitting/Receiving Serial Da

RXI is requested when the RDRF bit in SCSSR is set to 1.

ERI is requested when the ORER, PER, or FER bit in SCSSR is set to 1.

TEI is requested when the TEND bit in SCSSR is set to 1. Where the TXI interrupt indi
transmit data writing is enabled, the TEI interrupt indicates that the transmit operation i

Table 14.13 SCI Interrupt Sources

Interrupt Source	Description	Priority When Reset Is
ERI	Receive error (ORER, PER, or FER)	High
RXI	Receive data full (RDRF)	
TXI	Transmit data empty (TDRE)	
TEI	Transmit end (TEND)	Low

See section 4, Exception Handling, for priorities and the relationship to non-SCI interru

transmit data to SCTDR, be sure to check that TDRE is set to 1.

Simultaneous Multiple Receive Errors: Table 14.14 indicates the state of SCSSR st
when multiple receive errors occur simultaneously. When an overrun error occurs, the
contents cannot be transferred to SCRDR, so receive data is lost.

Table 14.14 SCSSR Status Flags and Transfer of Receive Data

	SCSSR Status Flags				Receive D
Receive Error Status	**RDRF**	**ORER**	**FER**	**PER**	**SCRSR →**
Overrun error	1	1	0	0	X
Framing error	0	0	1	0	O
Parity error	0	0	0	1	O
Overrun error + framing error	1	1	1	0	X
Overrun error + parity error	1	1	0	1	X
Framing error + parity error	0	0	1	1	O
Overrun error + framing error + parity error	1	1	1	1	X

X: Receive data is not transferred from SCRSR to SCRDR.
O: Receive data is transferred from SCRSR to SCRDR.

Break Detection and Processing: Break signals can be detected by reading the RxD
when a framing error (FER) is detected. In the break state, the input from the RxD pin
all 0s, so FER is set and the parity error flag (PER) may also be set. In the break state,
receiver continues to operate, so if the FER bit is cleared to 0, it will be set to 1 again.

Sending a Break Signal: The TxD pin I/O condition and level can be determined by
SCP0DT bit in the port SC data register (SCPDR) and bits SCP0MD0 and SCP0MD1
SC control register (SCPCR). This feature can be used to send breaks. To send a break
serial transmission, clear the SCP0DT bit to 0 (designating low level), then clear the T
(halting transmission). When the TE bit is cleared to 0, the transmitter is initialized reg
the current transmission state, and 0 is output from the TxD pin.

to 0 does not clear the receive error flags.

Receive Data Sampling Timing and Receive Margin in Asynchronous Mode: In as
mode, the SCI operates on a base clock of 16 times the transfer rate frequency. In recei
SCI synchronizes internally with the falling edge of the start bit, which it samples on th
clock. Receive data is latched at the rising edge of the eighth base clock pulse (figure 1

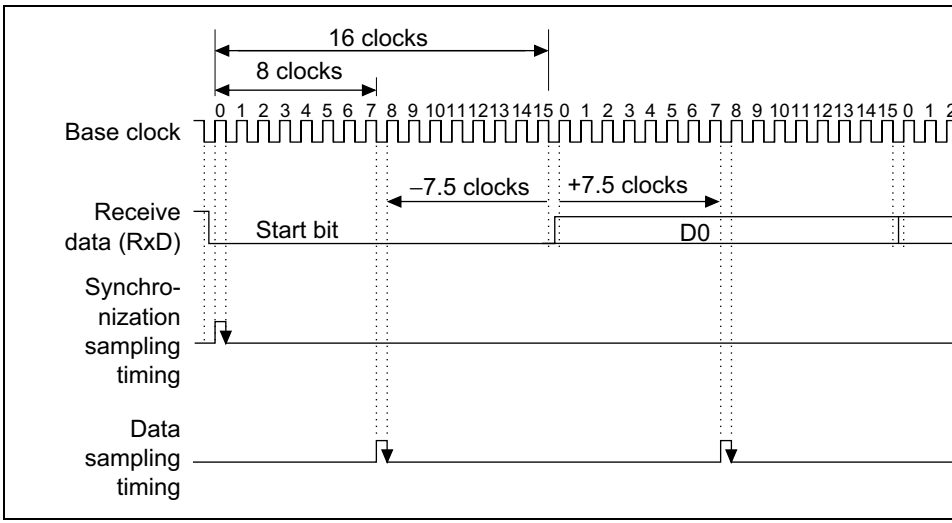


Figure 14.24 Receive Data Sampling Timing in Asynchronous Mode

L = Frame length (L = 9 to 12)
F = Absolute deviation of clock frequency

From equation 1, if F = 0 and D = 0.5, the receive margin is 46.875%, as in equation 2

Equation 2:

$$M = (0.5 - 1/(2 \times 16)) \times 100\%$$
$$= 46.875\%$$

This is a theoretical value. A reasonable margin to allow in system designs is 20% to 3

Notes on Synchronous External Clock Mode:

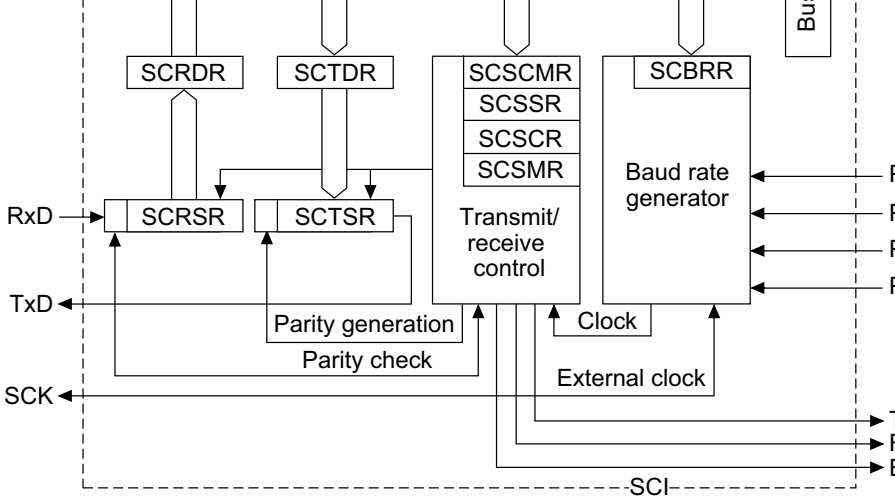
- Do not set TE = RE = 1 until at least four clocks after external clock SCK has char to 1.
- Set TE = RE = 1 only when external clock SCK is 1.
- When receiving, RDRF is set to 1 when RE is set to zero 2.5–3.5 clocks after the r the SCK input of the D7 bit in RxD, but data cannot be copied to SCRDR.

Note on Synchronous Internal Clock Mode: When receiving, RDRF is set to 1 when R to zero 1.5 clocks after the rising edge of the SCK output of the D7 bit in RxD, but da copied to SCRDR.

15.1.1 Features

The smart card interface has the following features:

- Asynchronous mode
  - Data length: 8 bits
  - Parity bit generation and check
  - Receive mode error signal detection (parity error)
  - Transmit mode error signal detection and automatic re-transmission of data
  - Supports both direct convention and inverse convention
- Bit rate can be selected using on-chip baud rate generator.
- Three types of interrupts: Transmit-data-empty, receive-data-full, and communicat
  interrupts are requested independently.



Legend
SCSCMR: Smart card mode register
SCRSR: Receive shift register
SCRDR: Receive data register
SCTSR: Transmit shift register
SCTDR: Transmit data register
SCSMR: Serial mode register
SCSCR: Serial control register
SCSSR: Serial status register
SCBRR: Bit rate register

Figure 15.1 Block Diagram of Smart Card Interface

Transmit data pin	TxD0	Output	Transmit data output

15.1.4 Smart Card Interface Registers

Table 15.2 summarizes the registers used by the smart card interface. The SCSMR, SC
SCSCR, SCTDR, and SCRDR registers are the same as for the normal SCI function. T
described in section 14, Serial Communication Interface (SCI).

Table 15.2 Registers

Name	Abbreviation	R/W	Initial Value*3	Address	A
Serial mode register	SCSMR	R/W	H'00	H'FFFFFE80	8
Bit rate register	SCBRR	R/W	H'FF	H'FFFFFE82	8
Serial control register	SCSCR	R/W	H'00	H'FFFFFE84	8
Transmit data register	SCTDR	R/W	H'FF	H'FFFFFE86	8
Serial status register	SCSSR	R/(W)*1	H'84	H'FFFFFE88	8
Receive data register	SCRDR	R	H'00	H'FFFFFE8A	8
Smart card mode register	SCSCMR	R/W	H'00*2	H'FFFFFE8C	8

Notes: 1. Only 0 can be written, to clear the flags.
2. Bits 0, 2, and 3 are cleared. The value of the other bits is undefined.
3. Initialized by a power-on or manual reset.

mode.

Bit:	7	6	5	4	3	2	1
	—	—	—	—	SDIR	SINV	—
Initial value:	—	—	—	—	0	0	—
R/W:	R	R	R	R	R/W	R/W	R

Bits 7 to 4 and 1—Reserved: These bits are always read as 0. The write value should a 0.

Bit 3—Smart Card Data Transfer Direction (SDIR): Selects the serial/parallel conv format.

Bit 3: SDIR	Description
0	Contents of SCTDR are transferred LSB-first, and receive data is SCRDR LSB-first (I
1	Contents of SCTDR are transferred MSB-first, and receive data is SCRDR MSB-first

Bit 2—Smart Card Data Inversion (SINV): Specifies whether to invert the logic leve data. This function is used in combination with bit 3 for transmitting and receiving with convention card. SINV does not affect the logic level of the parity bit. See section 15.3. Settings, for information on how parity is set.

Bit 2: SINV	Description
0	Contents of SCTDR are transferred unchanged, and receive data in SCRDR unchanged (I
1	Contents of SCTDR are inverted before transfer, and receive data inverted before storage in SCRDR

In smart card interface mode, the function of SCSSR bit 4 is changed. The setting con
bit 2, the TEND bit, are also changed.

Bit:	7	6	5	4	3	2	1
	TDRE	RDRF	ORER	FER/ERS	PER	TEND	MPE
Initial value:	1	0	0	0	0	1	0
R/W:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R	R

Note: Only 0 can be written, to clear the flag.

Bit 7—Transmit Data Register Empty (TDRE)
Bit 6—Receive Data Register Full (RDRE)
Bit 5—Overrun Error (ORER)
These bits have the same function as in the ordinary SCI. See section 14, Serial Comm
Interface (SCI), for more information.

Bit 4—Error Signal Status (ERS): In the smart card interface mode, bit 4 indicates t
the error signal returned from the receiving side during transmission. The smart card i
cannot detect framing errors.

Bit 4: ERS	Description
0	Receiving ended normally with no error signal
	[Clearing conditions]
	(1) By a reset or in standby mode
	(2) Cleared by reading ERS when ERS = 1, then writing 0 to ERS
1	An error signal indicating a parity error was transmitted from the rece
	[Setting condition]
	If the error signal sampled is low

Note: The ERS flag maintains its state even when the TE bit in SCSCR is cleared to

1	End of transmission [Setting conditions] (1) the chip is reset or enters standby mode, (2) the TE bit in SCSCR is 0 and the FER/ERS bit is also 0, (3) the C/$\overline{A}$ bit in SCSMR is 0, and TDRE = 1 and FER/ERS = 0 (norm transmission) 2.5 etu after a one-byte serial character is transmitte (4) the C/$\overline{A}$ bit in SCSMR is 1, and TDRE = 1 and FER/ERS = 0 (norm transmission) 1.0 etu after a one-byte serial character is transmitte	(

Note: etu: Elementary Time Unit (time for transfer of 1 bit).

15.3 Operation

15.3.1 Overview

The primary functions of the smart card interface are described below.

1. Each frame consists of 8-bit data and 1 parity bit.
2. During transmission, the card leaves a guard time of at least 2 etu (elementary time for transfer of 1 bit) from the end of the parity bit to the start of the next frame.
3. During reception, the card outputs an error signal low level for 1 etu after 10.5 etu h from the start bit if a parity error was detected.
4. During transmission, it automatically transmits the same data after allowing at least the time the error signal is sampled.
5. Only start-stop type asynchronous communication functions are supported; no sync communication functions are available.

on the IC card.

Use the chip's port output as the reset signal. Apart from these pins, power and ground connections are usually also required.

Note: When the IC card is not connected and both RE and TE are set to 1, closed co is possible and auto-diagnosis can be performed.

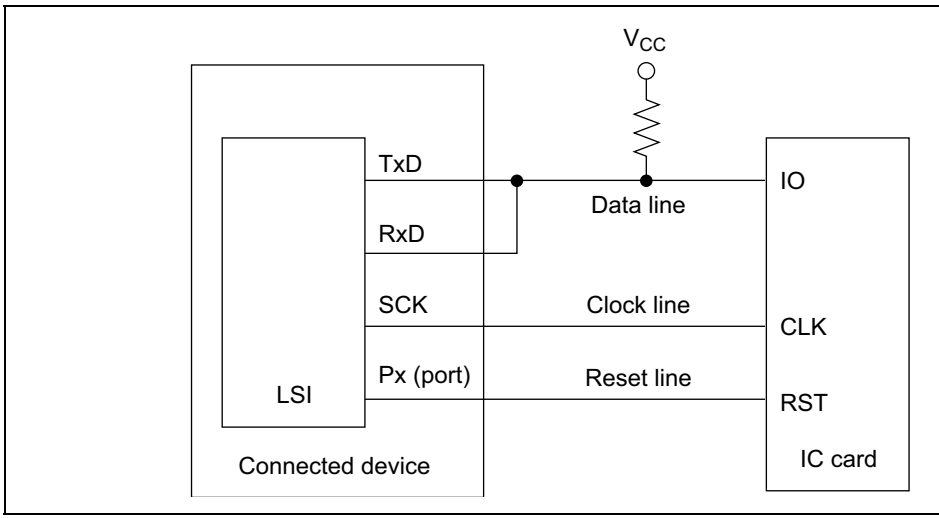


Figure 15.2 Pin Connection Diagram for Smart Card Interface

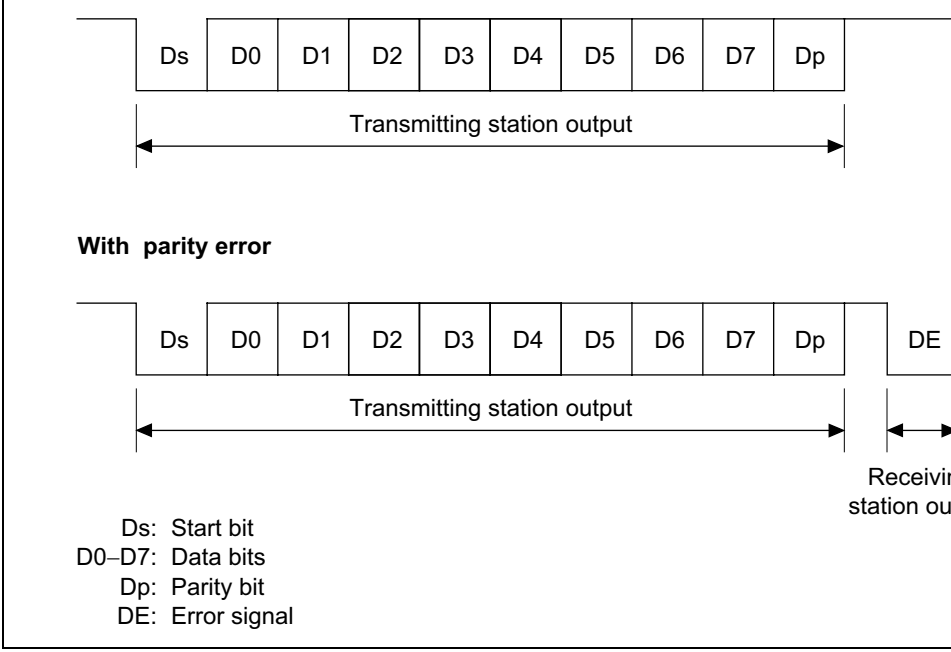


Figure 15.3 Data Format for Smart Card Interface

The operating sequence is:

1. The data line is high-impedance when not in use and is fixed high with a pull-up reg
2. The transmitting side starts one frame of data transmission. The data frame starts wi bit (Ds, low level). The start bit is followed by eight data bits (D0–D7) and a parity
3. On the smart card interface, the data line returns to high-impedance after this. The d pulled high with a pull-up register.
4. The receiving side checks parity. When the data is received normally with no parity receiving side then waits to receive the next data. When a parity error occurs, the re side outputs an error signal (DE, low level) and requests re-transfer of data. The rec station returns the signal line to high-impedance after outputting the error signal for period. The signal line is pulled high with a pull-up register.

Register	Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1
SCSMR	H'FFFFFE80	C/$\overline{A}$	0	1	O/$\overline{E}$	1	0	CKS
SCBRR	H'FFFFFE82	BRR7	BRR6	BRR5	BRR4	BRR3	BRR2	BRR
SCSCR	H'FFFFFE84	TIE	RIE	TE	RE	0	0	CKE
SCTDR	H'FFFFFE86	TDR7	TDR6	TDR5	TDR4	TDR3	TDR2	TDR
SCSSR	H'FFFFFE88	TDRE	RDRF	ORER	FER/ ERS	PER	TEND	0
SCRDR	H'FFFFFE8A	RDR7	RDR6	RDR5	RDR4	RDR3	RDR2	RDR
SCSCMR	H'FFFFFE8C	—	—	—	—	SDIR	SINV	—

Note: Dashes indicate unused bits.

1. Setting the serial mode register (SCSMR): The C/$\overline{A}$ bit selects the setting timing o flag, and selects the clock output state in combination with bits CKE1 and CKE0 i control register (SCSCR). Clear the O/$\overline{E}$ bit to 0 if the IC card uses the direct conv set it to 1 if the card uses the inverse convention. Select the on-chip baud rate gene source with the CKS1 and CKS0 bits (see section 15.3.5, Clock).
2. Setting the bit rate register (SCBRR): Set the bit rate. See section 15.3.5, Clock, to calculate the set value.
3. Setting the serial control register (SCSCR): The TIE, RIE, TE and RE bits functio for the ordinary SCI. See section 14, Serial Communication Interface (SCI), for m information. The CKE0 bit specifies the clock output. When no clock is output, cl 0; when a clock is output, set CKE0 to 1.
4. Setting the smart card mode register (SCSCMR): The SDIR and SINV bits are bot 0 for IC cards that use the direct convention, and both set to 1 when the inverse co used. The SMIF bit is set to 1 for the smart card interface.

   Figure 15.4 shows sample waveforms for register settings of the two types of IC c convention and inverse convention) and their start characters.

   In the direct convention type, the logical 1 level is state Z, the logical 0 level is sta communication is LSB-first. The start character data is H'3B. Parity is even (from card standard), and so the parity bit is 1.

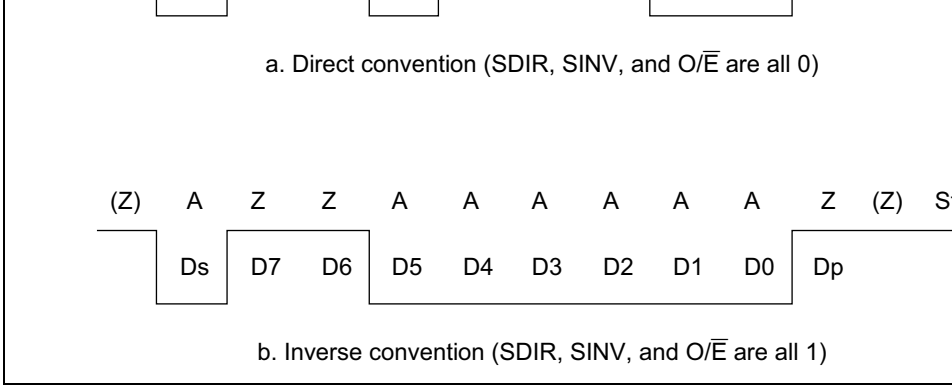


Figure 15.4 Waveform of Start Character

15.3.5 Clock

Only the internal clock generated by the on-chip baud rate generator can be used as the communication clock in the smart card interface. The bit rate for the clock is set by the register (SCBRR) and the CKS1 and CKS0 bits in the serial mode register (SCSMR), a calculated using the equation below. Table 15.5 shows sample bit rates. If clock output selected by setting CKE0 to 1, a clock with a frequency 372 times the bit rate is output SCK0 pin.

$$B = \frac{P\phi}{1488 \times 2^{2n-1} \times (N + 1)} \times 10^6$$

Where: N = Value set in SCBRR ($0 \leq N \leq 255$)
B = Bit rate (bits/s)
$P\phi$ = Peripheral module operating frequency (MHz)
n = 0 to 3 (table 15.4)

Table 15.5 Examples of Bit Rate B (Bits/s) for SCBRR Settings (n = 0)

	Pϕ (MHz)					
N	**7.1424**	**10.00**	**10.7136**	**13.00**	**14.2848**	**16.00**
0	9600.0	13440.9	14400.0	17473.1	19200.0	21505.4
1	4800.0	6720.4	7200.0	8736.6	9600.0	10752.7
2	3200.0	4480.3	4800.0	5824.4	6400.0	7168.5

Note: The bit rate is rounded to one decimal place.

Calculate the value to be set in the bit rate register (SCBRR) from the operating freque bit rate. N is an integer in the range $0 \leq N \leq 255$, specifying a smallish error.

$$N = \frac{P\phi}{1488 \times 2^{2n-1} \times B} \times 10^6 - 1$$

Table 15.6 Examples of SCBRR Settings for Bit Rate B (Bits/s) (n = 0)

	ϕ (MHz) (9600 Bits/s)											
7.1424		**10.00**		**10.7136**		**13.00**		**14.2848**		**16.00**	
N	**Error**	**N**	**Error**	**N**	**Error**	**N**	**Error**	**N**	**Error**	**N**	**Error**
0	0.00	1	30.00	1	25.00	1	8.99	1	0.00	1	12.01

16.00	21505	0	0
18.00	24194	0	0

The bit rate error is found as follows:

$$\text{Error (\%)} = \left(\frac{P\phi}{1488 \times 2^{2n-1} \times B \times (N + 1)} \times 10^6 - 1\right) \times 100$$

Table 15.8 shows the relationship between transmit/receive clock register set values and states on the smart card interface.

Table 15.8 Register Set Values and SCK Pin

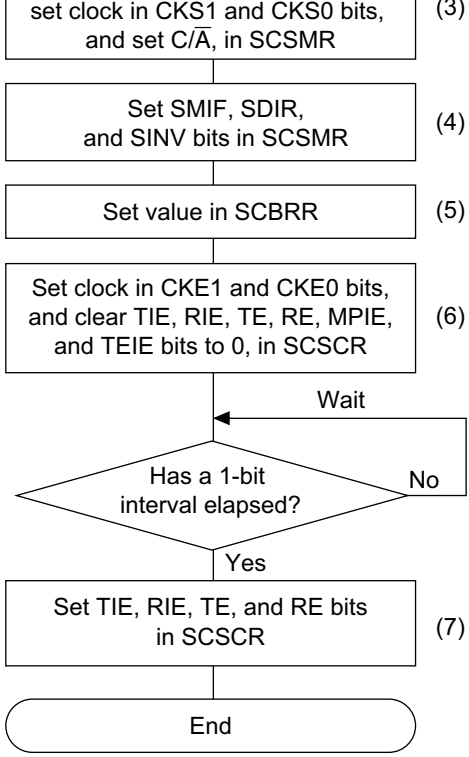
	Register Value				SCK Pin	
Setting	**SMIF**	**C/$\overline{A}$**	**CKE1**	**CKE0**	**Output**	**State**
1*1	1	0	0	0	Port	Determined by settin register SCP1MD1 a SCP1MD0 bits
	1	0	0	1	⎍⎍⎍	SCK (serial clock) ou
2*2	1	1	0	0	Low output	Low output state
	1	1	0	1	⎍⎍⎍	SCK (serial clock) ou
3*2	1	1	1	0	High output	High output state
	1	1	1	1	⎍⎍⎍	SCK (serial clock) ou

Notes: 1. The SCK output state changes as soon as the CKE0 bit is modified. The CK should be cleared to 0.

2. The clock duty remains constant despite stopping and starting of the clock b modification of the CKE0 bit.

in the serial mode register (SCSMR). At this time also clear the CHR and MP bits the STOP and PE bits to 1.

4. Set the SMIF, SDIR, and SINV bits in the smart card mode register (SCSCMR). W SMIF bit is set to 1, the TxD and RxD pins both switch from ports to SCI pins and high-impedance.
5. Set the value corresponding to the bit rate in the bit rate register (SCBRR).
6. Set the clock source select bits (CKE1 and CKE0 bits) in the serial control register Clear the TIE, RIE, TE, RE, MPIE, and TEIE bits to 0. When the CKE0 bit is set t is output from the SCK pin.
7. After waiting at least 1 bit, set the TIE, RIE, TE, and RE bits in SCSCR. Do not se RE bits simultaneously unless performing auto-diagnosis.

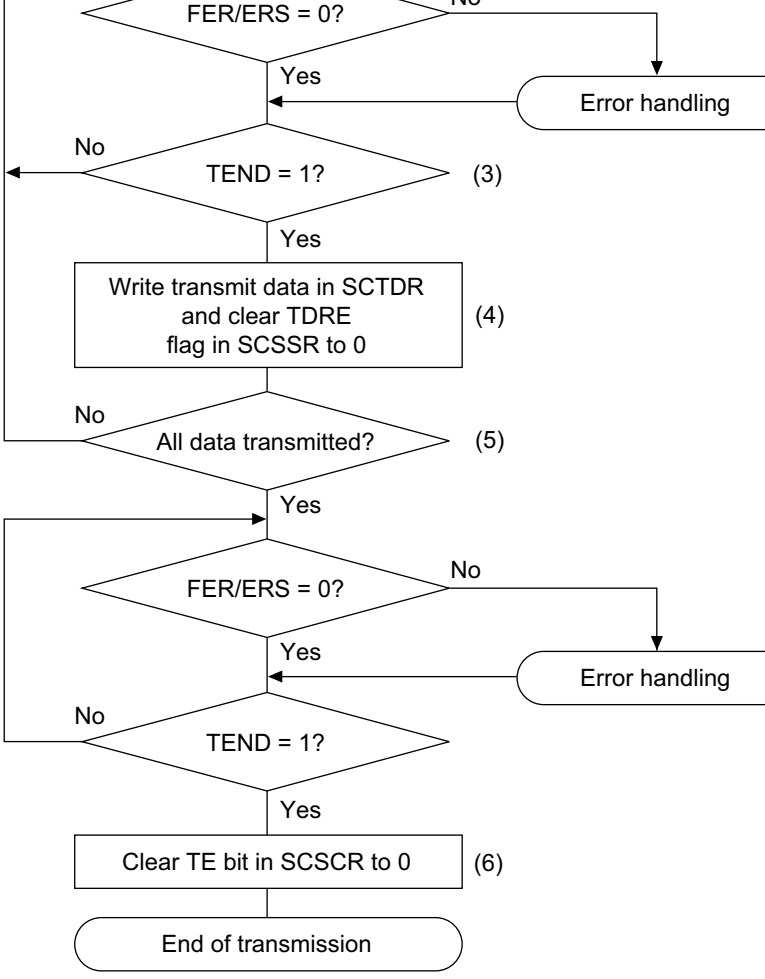


Note: Numbers in parentheses refer to steps in the preceding procedure description.

Figure 15.5 Initialization Flowchart (Example)

5. To transmit more data, return to step 2.
6. To end transmission, clear the TE bit to 0.

This processing can be interrupted. When the TIE bit is set to 1 and interrupt requests a transmit-data-empty interrupt (TXI) will be requested when the TEND flag is set to of transmission. When the RIE bit is set to 1 and interrupt requests are enabled, a com error interrupt (ERI) will be requested when the ERS flag is set to 1 when an error occ transmission. See Interrupt Operation below for more information.



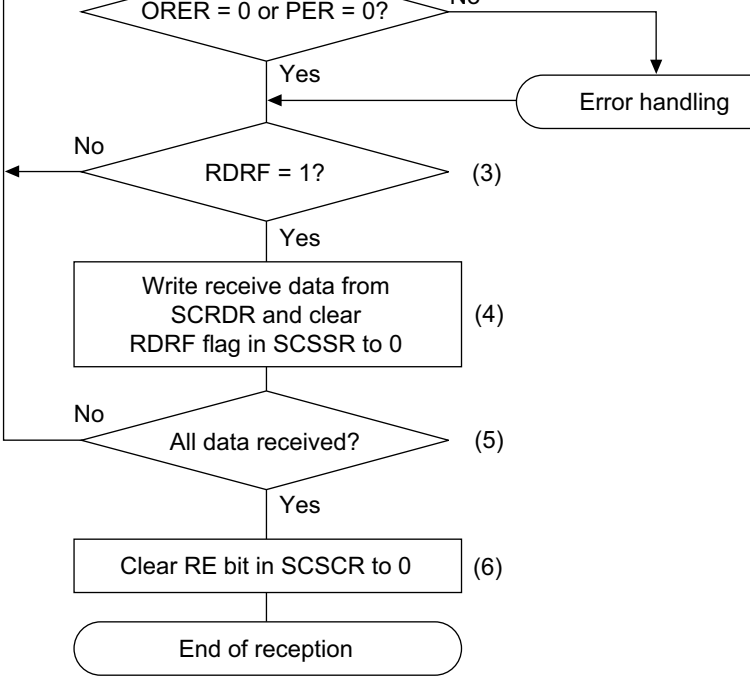
Note: Numbers in parentheses refer to steps in the preceding procedure description.

Figure 15.6 Transmission Flowchart

5. To receive more data, clear the RDRF flag to 0 and return to step 2.
6. To end reception, clear the RE bit to 0.

This processing can be interrupted. When the RIE bit is set to 1 and interrupt requests a receive-data-full interrupt (RXI) will be requested when the RDRF flag is set to 1 at reception. When an error occurs during reception and either the ORER or PER flag is communication error interrupt (ERI) will be requested. See Interrupt Operation below information.

The received data will be transferred to SCRDR even when a parity error occurs durin and PER is set to 1, so this data can still be read.



Note: Numbers in parentheses refer to steps in the preceding procedure description.

Figure 15.7 Reception Flowchart (Example)

the transmit-end interrupt (TEI) cannot be requested.

Set the TEND flag in SCSSR to 1 to request a TXI interrupt. Set the RDRF flag in SC request an RXI interrupt. Set the ORER, PER, or FER/ERS flag in SCSSR to 1 to requ interrupt (table 15.9).

Table 15.9 Smart Card Mode Operating State and Interrupt Sources

Mode	State	Flag	Mask Bit	Interru
Transmit mode	Normal	TEND	TIE	TXI
	Error	FER/ERS	RIE	ERI
Receive mode	Normal	RDRF	RIE	RXI
	Error	PER, ORER	RIE	ERI

15.4 Usage Notes

When the SCI is used as a smart card interface, be sure that all criteria in sections 15.4 Data Timing and Receive Margin in Asynchronous Mode and 15.4.2, Retransmission

15.4.1 Receive Data Timing and Receive Margin in Asynchronous Mode

In asynchronous mode, the SCI runs on a base clock with a frequency of 372 times the rate. During reception, the SCI samples the falling of the start bit using the base clock internal synchronization. Receive data is latched internally at the rising edge of the 18 clock cycle (figure 15.8).

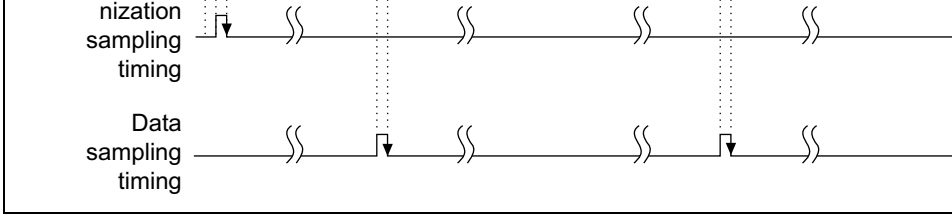


Figure 15.8 Receive Data Sampling Timing in Smart Card Mode

The receive margin is found from the following equation:

For smart card mode:

$$M = \left| (0.5 - \frac{1}{2N}) - (L - 0.5)F - \frac{|D - 0.5|}{N}(1 + F) \right| \times 100\%$$

Where: M = Receive margin (%)
N = Ratio of bit rate to clock (N = 372)
D = Clock duty (D = 0 to 1.0)
L = Frame length (L = 10)
F = Absolute value of clock frequency deviation

Using this equation, the receive margin when F = 0 and D = 0.5 is as follows:

$$M = (0.5 - 1/2 \times 372) \times 100\% = 49.866\%$$

3. When the received parity bit is checked and no error is found, the PER bit in SCSS
4. When the received parity bit is checked and no error is found, reception is conside been completed normally and the RDRF bit in SCSSR is automatically set to 1. If in SCSCR is enabled at this time, an RXI interrupt is requested.
5. When a normal frame is received, the pin maintains a three-state state when it tran error signal.

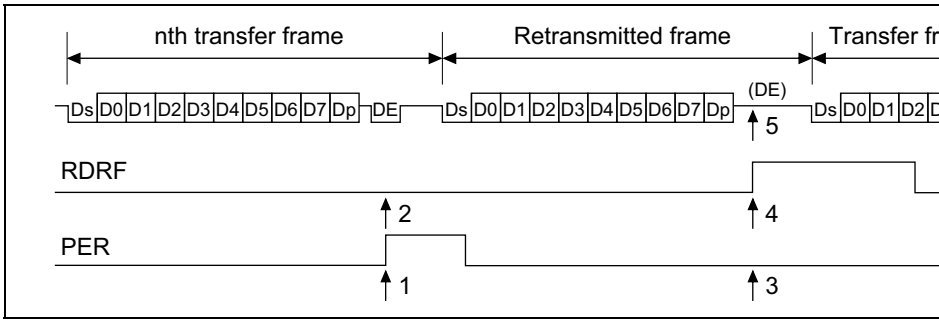


Figure 15.9 Retransmission in SCI Receive Mode

3. The FER/ERS bit in SCSSR is not set when no error signal is returned from the rece
4. When no error signal is returned from the receiving side, the TEND bit in SCSSR is when the transmission of the frame that includes the retransmission is considered co the TIE bit in SCSCR is enabled at this time, a TXI interrupt will be requested.

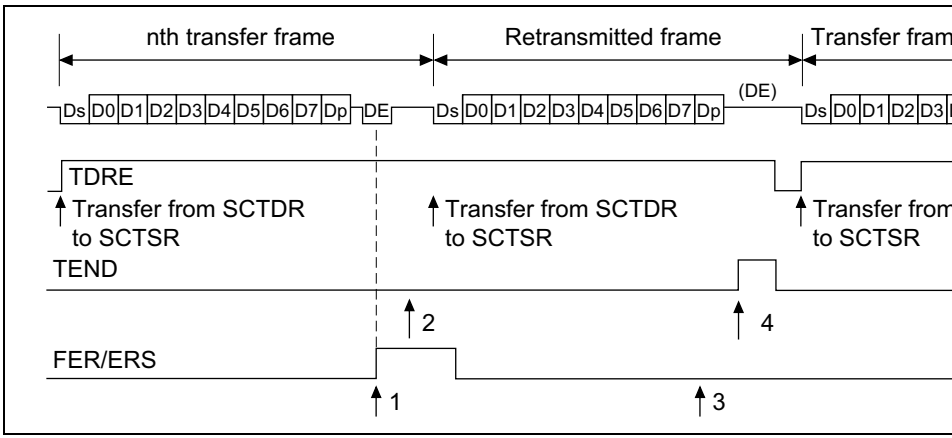
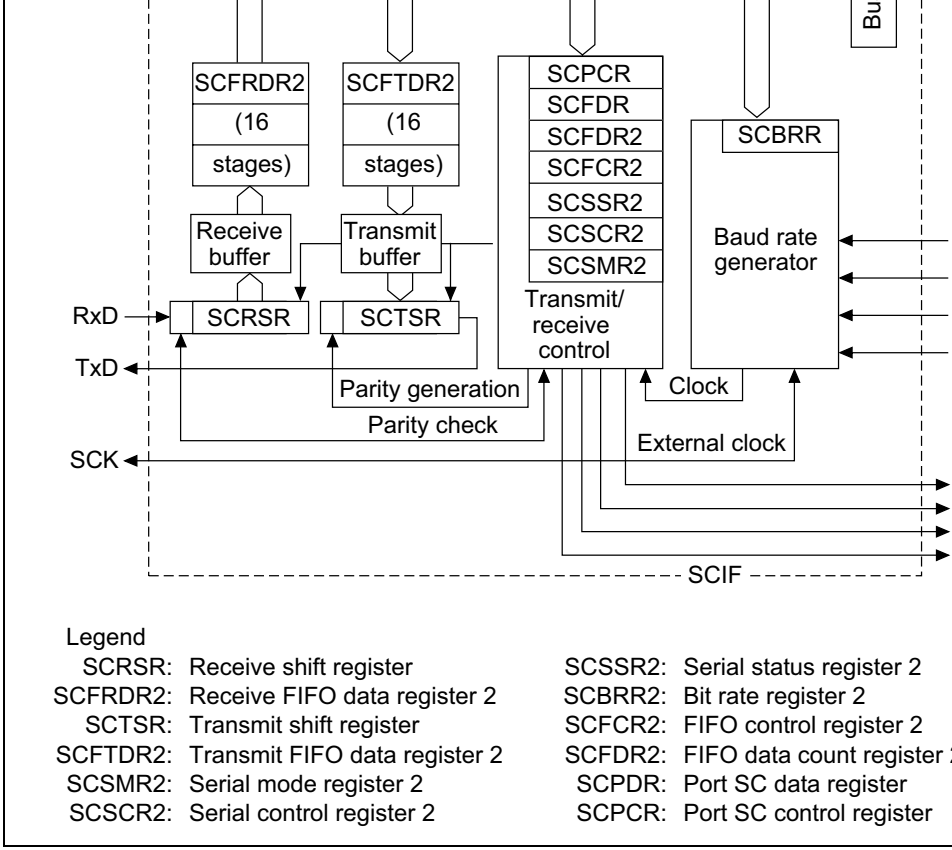


Figure 15.10 Retransmission in SCI Transmit Mode

reception that enable the SH7055 to perform efficient high-speed continuous commu

16.1.1 Features

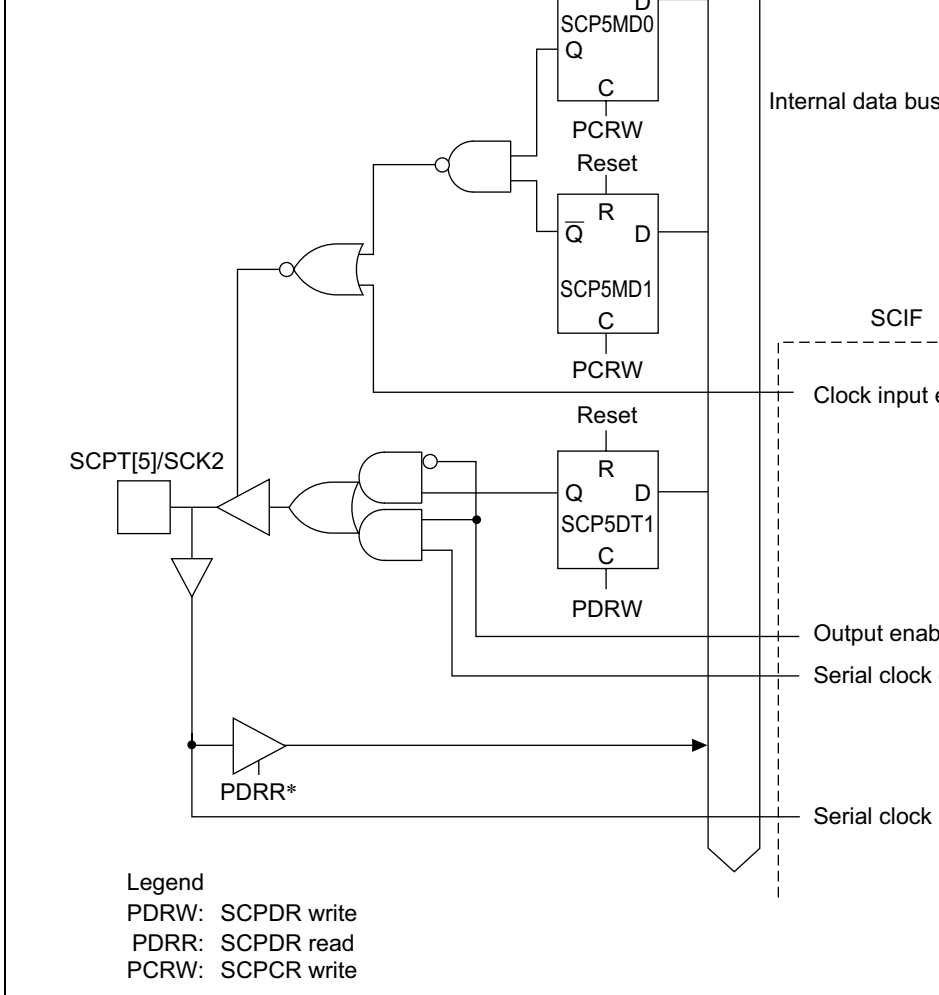
- Asynchronous serial communication:
  - Serial data communication is performed by start-stop in character units. The SC communicate with a universal asynchronous receiver/transmitter (UART), an a communication interface adapter (ACIA), or any other communications chip th a standard asynchronous serial system. There are eight selectable serial data communication formats.
  - Data length: 7 or 8 bits
  - Stop bit length: 1 or 2 bits
  - Parity: Even, odd, or none
  - Receive error detection: Parity and framing errors
  - Break detection: Break is detected when a framing error is followed by at least the space 0 level (low level). It is also detected by reading the RxD level direct port SC data register (SCPDR) when a framing error occurs.
- Full duplex communication: The transmitting and receiving sections are independe SCI can transmit and receive simultaneously. Both sections use 16-stage FIFO buf high-speed continuous data transfer is possible in both the transmit and receive dir
- On-chip baud rate generator with selectable bit rates
- Internal or external transmit/receive clock source: From either baud rate generator SCK pin (external)
- Four types of interrupts: Transmit-FIFO-data-empty, break, receive-FIFO-data-ful receive-error interrupts are requested independently. The direct memory access co (DMAC) can be activated to execute a data transfer by a transmit-FIFO-data-empt FIFO-data-full interrupt.
- When the SCIF is not in use, it can be stopped by halting the clock supplied to it, s power.
- On-chip modem control functions (RTS and CTS)
- The quantity of data in the transmit and receive FIFO registers and the number of r errors of the receive data in the receive FIFO register can be ascertained.
- A time-out error (DR) can be detected when receiving.



Legend

SCRSR:	Receive shift register	SCSSR2:	Serial status register 2
SCFRDR2:	Receive FIFO data register 2	SCBRR2:	Bit rate register 2
SCTSR:	Transmit shift register	SCFCR2:	FIFO control register 2
SCFTDR2:	Transmit FIFO data register 2	SCFDR2:	FIFO data count register 2
SCSMR2:	Serial mode register 2	SCPDR:	Port SC data register
SCSCR2:	Serial control register 2	SCPCR:	Port SC control register

Figure 16.1 Block Diagram of SCIF



Note: * When reading the SCK2 pin, clear the CKE1 and CKE0 bits in SCSCR to 0, and set the SCP5MD1 bit in SCSPR to 1 (see sectio SC Port Control Register (SCPCR)/SC Port Data Register (SCPDR)).

Figure 16.2 SCPT[5]/SCK2 Pin

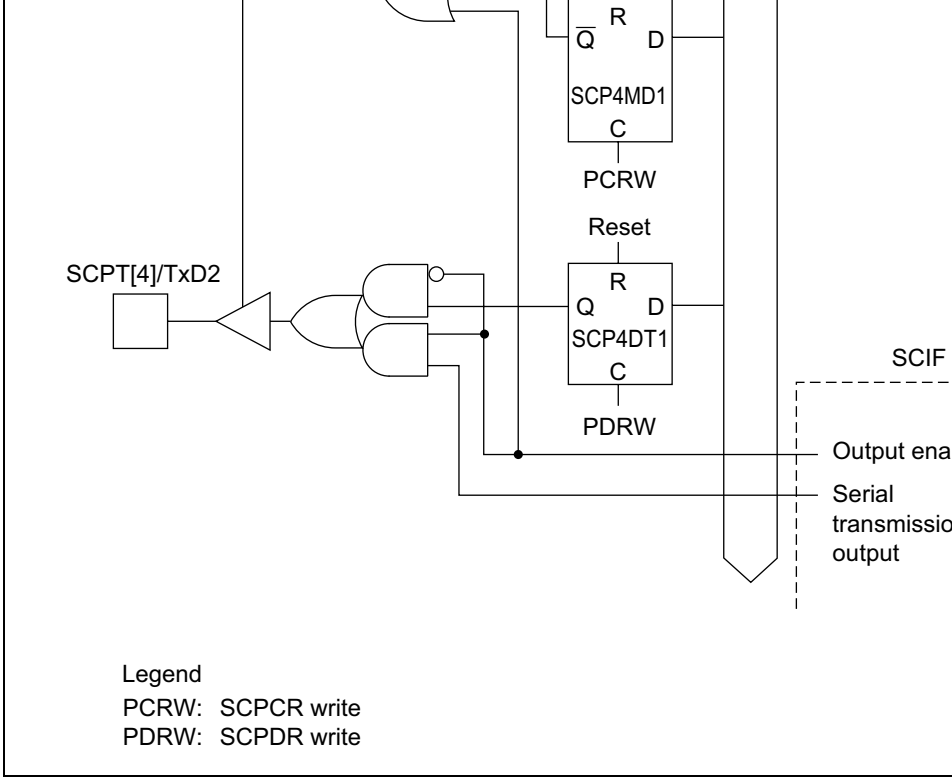


Figure 16.3 SCPT[4]/TxD2 Pin

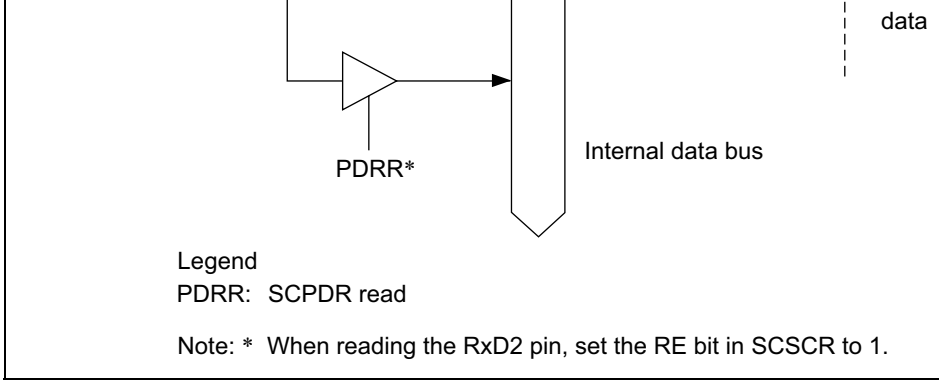


Figure 16.4 SCPT[4]/RxD2 Pin

16.1.3 Pin Configuration

The SCIF has the serial pins summarized in table 16.1.

Table 16.1 SCIF Pins

Pin Name	Abbreviation	I/O	Function
Serial clock pin	SCK2	I/O	Clock I/O
Receive data pin	RxD2	Input	Receive data input
Transmit data pin	TxD2	Output	Transmit data outp
Request to send pin	$\overline{\text{RTS2}}$	Output	Request to send
Clear to send pin	$\overline{\text{CTS2}}$	Input	Clear to send

Bit rate register 2	SCBRR2	R/W	H'FF	H'04000152 (H'A4000152)*2	8
Serial control register 2	SCSCR2	R/W	H'00	H'04000154 (H'A4000154)*2	8
Transmit FIFO data register 2	SCFTDR2	W	—	H'04000156 (H'A4000156)*2	8
Serial status register 2	SCSSR2	R/(W)*1	H'0060	H'04000158 (H'A4000158)*2	1
Receive FIFO data register 2	SCFRDR2	R	Undefined	H'0400015A (H'A400015A)*2	8
FIFO control register 2	SCFCR2	R/W	H'00	H'0400015C (H'A400015C)*2	8
FIFO data count register 2	SCFDR2	R	H'0000	H'0400015E (H'A400015E)*2	1

Notes: These registers are located in area 1 of physical space. Therefore, when the ca
either access these registers from the P2 area of logical space or else make an
setting using the MMU so that these registers are not cached.

1. Only 0 can be written to clear the flag.
2. When address translation by the MMU does not apply, the address in parent
should be used.

Bit:	7	6	5	4	3	2	1
R/W:	—	—	—	—	—	—	—

16.2.2 Receive FIFO Data Register (SCFRDR)

The 16-byte receive FIFO data register (SCFRDR) stores serial receive data. The SCI the reception of one byte of serial data by moving the received data from the receive s (SCRSR) into SCFRDR for storage. Continuous reception is possible until 16 bytes ar

The CPU can read but not write to SCFRDR. If data is read when there is no receive d SCFRDR, the value is undefined. When this register is full of receive data, subsequent is lost.

Bit:	7	6	5	4	3	2	1
R/W:	R	R	R	R	R	R	R

16.2.3 Transmit Shift Register (SCTSR)

The transmit shift register (SCTSR) transmits serial data. The SCI loads transmit data transmit FIFO data register (SCFTDR) into SCTSR, then transmits the data serially fr pin, LSB (bit 0) first. After transmitting one data byte, the SCI automatically loads the transmit data from SCFTDR into SCTSR and starts transmitting again. The CPU cann write to SCTSR directly.

Bit:	7	6	5	4	3	2	1
R/W:	—	—	—	—	—	—	—

data is attempted, the data is ignored.

Bit:	7	6	5	4	3	2	1
R/W:	W	W	W	W	W	W	W

16.2.5 Serial Mode Register (SCSMR)

The serial mode register (SCSMR) is an 8-bit register that specifies the SCIF serial communication format and selects the clock source for the baud rate generator.

The CPU can always read and write to SCSMR. SCSMR is initialized to H'00 by a rese standby or module standby mode.

Bit:	7	6	5	4	3	2	1
	—	CHR	PE	O/$\overline{E}$	STOP	—	CKS1
Initial value:	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R	R/W

Bit 7—Reserved: This bit is always read as 0. The write value should always be 0.

Bit 6—Character Length (CHR): Selects 7-bit or 8-bit data in asynchronous mode.

Bit 6: CHR	Description
0	8-bit data (Ir
1	7-bit data*

Note: * When 7-bit data is selected, the MSB (bit 7) of the transmit FIFO data register i transmitted.

Bit 4—Parity Mode (O/$\overline{E}$): Selects even or odd parity when parity bits are added and
The O/$\overline{E}$ setting is used only when the parity enable bit (PE) is set to 1 to enable parity
and checking. The O/$\overline{E}$ setting is ignored when parity addition and checking is disable

Bit 4: O/$\overline{E}$	Description	
0	Even parity*1	(
1	Odd parity*2	

Notes: 1. If even parity is selected, the parity bit is added to transmit data to make an
number of 1s in the transmitted character and parity bit combined. Receive
checked to see if it has an even number of 1s in the received character and
combined.
2. If odd parity is selected, the parity bit is added to transmit data to make an
of 1s in the transmitted character and parity bit combined. Receive data is c
see if it has an odd number of 1s in the received character and parity bit co

Bit 3—Stop Bit Length (STOP): Selects one or two bits as the stop bit length.

When receiving, only the first stop bit is checked, regardless of the STOP bit setting. I
stop bit is 1, it is treated as a stop bit, but if the second stop bit is 0, it is treated as the
the next incoming character.

Bit 3: STOP	Description	
0	One stop bit*1	(
1	Two stop bits*2	

Notes: 1. When transmitting, a single 1-bit is added at the end of each transmitted ch
2. When transmitting, two 1-bits are added at the end of each transmitted cha

Bit 2—Reserved: This bit is always read as 0. The write value should always be 0.

1	Pϕ/64

Note: Pϕ: Peripheral clock

16.2.6 Serial Control Register (SCSCR)

The serial control register (SCSCR) operates the SCIF transmitter/receiver, selects the s output in asynchronous mode, enables/disables interrupt requests, and selects the transm clock source. The CPU can always read and write to SCSCR. SCSCR is initialized to H reset and in standby or module standby mode.

Bit:	7	6	5	4	3	2	1
	TIE	RIE	TE	RE	—	—	CKE1
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R	R	R/W

Bit 7—Transmit Interrupt Enable (TIE): Enables or disables the transmit-FIFO-data interrupt (TXI) requested when the serial transmit data is transferred from the transmit register (SCFTDR) to the transmit shift register (SCTSR), when the quantity of data in transmit FIFO register becomes less than the specified number of transmission triggers, the TDFE flag in the serial FIFO status register (SCFSR) is set to1.

Bit 7: TIE	Description
0	Transmit-FIFO-data-empty interrupt request (TXI) is disabled* (In
1	Transmit-FIFO-data-empty interrupt request (TXI) is enabled

Note: * The TXI interrupt request can be cleared by writing a greater quantity of transm the specified transmission trigger number to SCFTDR and by clearing TDFE to reading 1 from TDFE, or can be cleared by clearing TIE to 0.

1	Receive-data-full interrupt (RXI) and receive-error interrupt (ERI) requ enabled

Note: * RXI and ERI interrupt requests can be cleared by reading the DR, ER, or RDF has been set to 1, then clearing the flag to 0, or by clearing RIE to 0. With the read 1 from the RDF flag and clear it to 0, after reading receive data from SCF quantity of receive data becomes less than the specified receive trigger numb

Bit 5—Transmit Enable (TE): Enables or disables the SCIF serial transmitter.

Bit 5: TE	Description
0	Transmitter disabled
1	Transmitter enabled*

Note: * Serial transmission starts after writing of transmit data into SCFTDR2. Select format in SCSMR2 and SCFCR2 and reset the TFIFO before setting TE to 1.

Bit 4—Receive Enable (RE): Enables or disables the SCIF serial receiver.

Bit 4: RE	Description
0	Receiver disabled*1
1	Receiver enabled*2

Notes: 1. Clearing RE to 0 does not affect the receive flags (DR, ER, BRK, FER, PER ORER). These flags retain their previous values.

2. Serial reception starts when a start bit is detected. Select the receive forma SCSMR2 before setting RE to 1.

Bits 3 and 2—Reserved: These bits are always read as 0. The write value should alw

Bit 1: CKE1	Bit 0: CKE0	Description
0	0	Internal clock, SCK pin used for input pin (input signa ignored) (In
	1	Internal clock, SCK pin used for clock output*1
1	0	External clock, SCK pin used for clock input*2
	1	External clock, SCK pin used for clock input*2

Notes: 1. The output clock frequency is 16 times the bit rate.
2. The input clock frequency is 16 times the bit rate.

16.2.7 Serial Status Register (SCSSR)

The serial status register (SCSSR) is a 16-bit register. The upper 8 bits indicate the nur receive errors in the SCFRDR data, and the lower 8 bits indicate the SCIF operating sta

The CPU can always read and write to SCSSR, but cannot write 1 to the status flags (E TDFE, BRK, OPER, and DR). These flags can be cleared to 0 only if they have first be (after being set to 1). Bits 3 (FER) and 2 (PER) are read-only bits that cannot be written initialized to H'0060 by a reset and in standby or module standby mode.

Lower 8 bits:	7	6	5	4	3	2	1
	ER	TEND	TDFE	BRK	FER	PER	RDF
Initial value:	0	1	1	0	0	0	0
R/W:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R	R	R/(W)*

Note: * The only value that can be written is 0 to clear the flag.

	(2) When 0 is written after 1 is read from ER
1	A framing error or parity error has occurred[*2] [Setting conditions] (1) ER is set to 1 when the stop bit is 0 after checking whether or no stop bit of the received data is 1 at the end of one data receive o (2) When the total number of 1s in the receive data plus parity bit do the even/odd parity specified by the O/$\overline{E}$ bit in SCSMR

Notes: 1. Clearing the RE bit to 0 in SCSCR does not affect the ER bit, which retains value. Even if a receive error occurs, the receive data is transferred to SCF the receive operation is continued. Whether or not the data read from SCR a receive error can be detected by the FER and PER bits in SCSSR.

2. In stop mode, only the first stop bit is checked; the second stop bit is not ch

Bit 6—Transmit End (TEND): Indicates that when the last bit of a serial character w transmitted, SCFTDR did not contain valid data, so transmission has ended.

Bit 6: TEND	Description
0	Transmission is in progress [Clearing condition] When data is written in SCFTDR
1	End of transmission ([Setting conditions] (1) When the chip is reset or enters standby mode, when TE is clear the serial control register (SCSCR) (2) When SCFTDR does not contain receive data when the last bit o serial character is transmitted

	[Clearing condition] TDFE is cleared to 0 when data exceeding the specified transmission number is written to SCFTDR, or when software reads TDFE after it h set to 1, then writes 0 to TDFE
1	The quantity of transmit data in SCFTDR is less than the specified tra trigger number* [Setting conditions] (1) TDFE is set to 1 by a reset or in standby mode (2) When the quantity of transmit data in SCFTDR becomes less than specified transmission trigger number as a result of transmission

Note: * Since SCFTDR is a 16-byte FIFO register, the maximum quantity of data that c written when TDFE is 1 is “16 minus the specified transmission trigger number” attempt is made to write additional data, the data is ignored. The quantity of da SCFTDR is indicated by the upper 8 bits of SCFTDR.

Bit 4—Break Detection (BRK): Indicates that a break signal has been detected in rece

Bit 4: BRK	Description
0	No break signal received (In [Clearing conditions] (1) BRK is cleared to 0 when the chip is reset or enters standby mode (2) When software reads BRK after it has been set to 1, then writes 0
1	Break signal received* [Setting conditions] (1) BRK is set to 1 when data including a framing error is received (2) A framing error occurs with space 0 in the subsequent receive dat

Note: * When a break is detected, transfer of the receive data (H'00) to SCFRDR stops detection. When the break ends and the receive signal becomes mark 1, the tr receive data resumes. The receive data of a frame in which a break signal is d transferred to SCFRDR. After this, however, no receive data is transferred unti ends with the received signal being mark 1, and the next data is received.

1	A receive framing error occurred in the data read from SCFRDR [Setting condition] When a framing error is present in the data read from SCFRDR

Bit 2—Parity Error (PER): Indicates a parity error in the data read from the receive register (SCFRDR).

Bit 2: PER	Description
0	No receive parity error occurred in the data read from SCFRDR ([Clearing conditions] (1) When the chip undergoes a power-on reset or enters standby mo (2) When no parity error is present in the data read from SCFRDR
1	A receive framing error occurred in the data read from SCFRDR [Setting condition] When a parity error is present in the data read from SCFRDR

	(1) By a power-on reset or in standby mode (2) When the quantity of receive data in SCFRDR is less than the spe receive trigger value and 1 is read from RDF, which is then cleare
1	The quantity of receive data in SCFRDR is greater than the specified trigger number [Setting condition] When a quantity of receive data greater than the specified receive trig is stored in SCFRDR*

Note: * Since SCFTDR is a 16-byte FIFO register, the maximum quantity of data that c when RDF is 1 is the specified receive trigger number. If an attempt is made to all the data in SCFRDR has been read, the data is undefined. The quantity of r in SCFRDR is indicated by the lower 8 bits of SCFTDR.

Bit 0—Receive Data Ready (DR): Indicates that the quantity of data in the receive FI register (SCFRDR) is less than the specified receive trigger number, and that the next d yet been received after the elapse of 15 etu from the last stop bit.

Bit 0: DR	Description
0	Receiving is in progress, or no receive data remains in SCFRDR after ended normally (I [Clearing conditions] (1) When the chip undergoes a power-on reset or enters standby mo (2) When software reads DR after it has been set to 1, then writes 0 t
1	Next receive data has not been received [Setting condition] When SCFRDR contains less data than the specified receive trigger n and the next data has not yet been received after the elapse of 15 etu last stop bit*

Note: * This is equivalent to 1.5 frames with the 8-bit, 1-stop-bit format. (etu: elementar

Bits 11 to 8—Number of Framing Errors 3 to 0 (FER3 to FER0): Indicate the qua
including a framing error in the receive data stored in SCFRDR. The value indicated
8 represents the number of framing errors in SCFRDR.

16.2.8 Bit Rate Register (SCBRR)

The bit rate register (SCBRR) is an 8-bit register that, together with the baud rate gene
source selected by the CKS1 and CKS0 bits in the serial mode register (SCSMR), dete
serial transmit/receive bit rate.

The CPU can always read and write to SCBRR. SCBRR is initialized to H'FF by a res
module standby or standby mode. Each channel has independent baud rate generator c
different values can be set in two channels.

Bit:	7	6	5	4	3	2	1
Initial value:	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The SCBRR setting is calculated as follows:

Asynchronous mode:

$$N = \frac{P\phi}{64 \times 2^{2n-1} \times B} \times 10^6 - 1$$

B: Bit rate (bits/s)
N: SCBRR setting for baud rate generator ($0 \leq N \leq 255$)
$P\phi$: Operating frequency for peripheral modules (MHz)
n: Baud rate generator clock source (n = 0, 1, 2, 3) (for the clock sources ar
n, see table 16.3.)

Note: The bit rate error is given by the following formula:

$$\text{Error (\%)} = \left\{ \frac{P\phi}{(N+1) \times 64 \times 2^{2n-1} \times B} \times 10^6 - 1 \right\} \times 100$$

Table 16.4 lists examples of SCBRR settings.

Table 16.4 Bit Rates and SCBRR Settings

	Pϕ (MHz)							
	2			2.097152			2.457	
Bit Rate (bits/s)	n	N	Error (%)	n	N	Error (%)	n	N
110	1	141	0.03	1	148	–0.04	1	174
150	1	103	0.16	1	108	0.21	1	127
300	0	207	0.16	0	217	0.21	0	255
600	0	103	0.16	0	108	0.21	0	127
1200	0	51	0.16	0	54	–0.70	0	63
2400	0	25	0.16	0	26	1.14	0	31
4800	0	12	0.16	0	13	–2.48	0	15
9600	0	6	–6.99	0	6	–2.48	0	7
19200	0	2	8.51	0	2	13.78	0	3
31250	0	1	0.00	0	1	4.86	0	1
38400	0	1	–18.62	0	0	–14.67	0	1

2400	0	38	0.16	0	47	0.00	0	51
4800	0	19	–2.34	0	23	0.00	0	25
9600	0	9	–2.34	0	11	0.00	0	12
19200	0	4	–2.34	0	5	0.00	0	6
31250	0	2	0.00	0	3	–7.84	0	3
38400	—	—	—	0	2	0.00	0	2

	Pϕ (MHz)							
	4.9152			5			6	
Bit Rate (bits/s)	**n**	**N**	**Error (%)**	**n**	**N**	**Error (%)**	**n**	**N**
110	2	86	0.31	2	88	–0.25	2	106
150	1	255	0.00	2	64	0.16	2	77
300	1	127	0.00	1	129	0.16	1	155
600	0	255	0.00	1	64	0.16	1	77
1200	0	127	0.00	0	129	0.16	0	155
2400	0	63	0.00	0	64	0.16	0	77
4800	0	31	0.00	0	32	–1.36	0	38
9600	0	15	0.00	0	15	1.73	0	19
19200	0	7	0.00	0	7	1.73	0	9
31250	0	4	–1.70	0	4	0.00	0	5
38400	0	3	0.00	0	3	1.73	0	4

2400	0	79	0.00	0	95	0.00	0	103
4800	0	39	0.00	0	47	0.00	0	51
9600	0	19	0.00	0	23	0.00	0	25
19200	0	9	0.00	0	11	0.00	0	12
31250	0	5	2.40	0	6	5.33	0	7
38400	0	4	0.00	0	5	0.00	0	6

	Pϕ (MHz)									
	9.8304			10			12			12
Bit Rate (bits/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N
110	1	174	–0.26	2	177	–0.25	1	212	0.03	2	21
150	1	127	0.00	2	129	0.16	1	155	0.16	2	15
300	0	255	0.00	2	64	0.16	1	77	0.16	2	79
600	0	127	0.00	1	129	0.16	0	155	0.16	1	15
1200	0	255	0.00	1	64	0.16	0	77	0.16	1	79
2400	0	127	0.00	0	129	0.16	0	38	0.16	0	15
4800	0	63	0.00	0	64	0.16	0	19	0.16	0	79
9600	0	31	0.00	0	32	–1.36	0	9	0.16	0	39
19200	0	15	0.00	0	15	1.73	0	4	0.16	0	19
31250	0	9	–1.70	0	9	0.00	0	2	0.00	0	11
38400	0	1	0.00	0	7	1.73	0	9	–2.34	0	9

1200	1	95	0.00	1	103	0.16	1	127	0.00	1	1
2400	0	191	0.00	0	207	0.16	0	255	0.00	0	6
4800	0	95	0.00	0	103	0.16	0	127	0.00	0	1
9600	0	47	0.00	0	51	0.16	0	63	0.00	0	6
19200	0	23	0.00	0	25	0.16	0	31	0.00	0	3
31250	0	14	–1.70	0	15	0.00	0	19	–1.70	0	1
38400	0	11	0.00	0	12	0.16	0	15	0.00	0	1
115200	0	3	0.00	0	3	8.51	0	4	6.67	0	4
500000	0	0	–7.84	0	0	0.00	0	0	22.9	0	0

	Pϕ (MHz)										
	24			24.576			28.7				
Bit Rate (bits/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N
110	3	106	–0.44	3	108	0.08	3	126	0.31	3	1
150	3	77	0.16	3	79	0.00	3	92	0.46	3	9
300	2	155	0.16	2	159	0.00	2	186	–0.08	2	1
600	2	77	0.16	2	79	0.00	2	92	0.46	2	9
1200	1	155	0.16	1	159	0.00	1	186	–0.08	1	1
2400	1	77	0.16	1	79	0.00	1	92	0.46	1	9
4800	0	155	0.16	0	159	0.00	0	186	–0.08	0	1
9600	0	77	0.16	0	79	0.00	0	92	0.46	0	9
19200	0	38	0.16	0	39	0.00	0	46	–0.61	0	4
31250	0	23	0.00	0	24	–1.70	0	28	–1.03	0	2
38400	0	19	–2.34	0	19	0.00	0	22	1.55	0	2
115200	0	6	–6.99	0	6	–4.76	0	7	–2.68	0	7
500000	0	1	–25.0	0	1	–23.2	0	1	–10.3	0	1

2.097152	65536	0	0
2.4576	76800	0	0
3	93750	0	0
3.6864	115200	0	0
4	125000	0	0
4.9152	153600	0	0
8	250000	0	0
9.8304	307200	0	0
12	375000	0	0
14.7456	460800	0	0
16	500000	0	0
19.6608	614400	0	0
20	625000	0	0
24	750000	0	0
24.576	768000	0	0
28.7	896875	0	0
30	937500	0	0

4	1.0000	62500
4.9152	1.2288	76800
8	2.0000	125000
9.8304	2.4576	153600
12	3.0000	187500
14.7456	3.6864	230400
16	4.0000	250000
19.6608	4.9152	307200
20	5.0000	312500
24	6.0000	375000
24.576	6.1440	384000
28.7	7.1750	448436
30	7.5000	468750

registers, sets the trigger data quantity, and contains an enable bit for loop-back testing. can always be read and written to by the CPU. It is initialized to H'00 by a reset, by the standby function, and in standby mode.

Bits 7 and 6—Receive FIFO Data Trigger (RTRG1, RTRG0): Set the quantity of re which sets the receive data full (RDF) flag in the serial status register (SCSSR). The RD set to 1 when the quantity of receive data stored in the receive FIFO register (SCFRDR) the set trigger number shown below.

Bit 7: RTRG1	Bit 6: RTRG0	Receive Trigger Number
0	0	1 (In
0	1	4
1	0	8
1	1	14

Bits 5 and 4—Transmit FIFO Data Trigger (TTRG1, TTRG0): Set the quantity of t transmit data which sets the transmit FIFO data register empty (TDFE) flag in the serial register (SCSSR). The TDFE flag is set to 1 when the quantity of transmit data in the tr FIFO data register (SCFTDR) becomes less than the set trigger number shown below.

Bit 5: TTRG1	Bit 4: TTRG0	Transmit Trigger Number
0	0	8 (8)*
0	1	4 (12)
1	0	2 (14)
1	1	1 (15)

Note: * Initial value. Values in parentheses mean the number of empty bits in SCFTDR TDFE flag is set to 1.

FIFO data register and resets the data to the empty state.

Bit 2: TFRST	Description	
0	Reset operation disabled*	(
1	Reset operation enabled	

Note: * Reset is executed in a reset or in standby mode.

Bit 1—Receive FIFO Data Register Reset (RFRST): Disables the receive data in th FIFO data register and resets the data to the empty state.

Bit 1: RFRST	Description	
0	Reset operation disabled*	(
1	Reset operation enabled	

Note: * Reset is executed in a reset or in standby mode.

Bit 0—Loop-Back Test (LOOP): Internally connects the transmit output pin (TXD) input pin (RXD) and enables loop-back testing.

Bit 0: LOOP	Description	
0	Loop back test disabled	(
1	Loop back test enabled	

Initial value:	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R

The upper 8 bits of SCFDR indicate the quantity of non-transmitted data stored in SCF means no transmit data, and H'10 means that SCFTDR is full of transmit data.

Lower 8 Bits:	7	6	5	4	3	2	1
	—	—	—	R4	R3	R2	R1
Initial value:	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R

The lower 8 bits of SCFDR indicate the quantity of receive data stored in SCFRDR. H' no receive data, and H'10 means that SCFRDR full of receive data.

(SCSMR), as shown in table 16.7. The SCIF clock source is selected by the combinati
CKE1 and CKE0 bits in the serial control register (SCSCR), as shown in table 16.8.

- Data length is selectable: 7 or 8 bits.
- Parity and multiprocessor bits are selectable, as is the stop bit length (1 or 2 bits). T
combination of the preceding selections constitutes the communication format and
length.
- In receiving, it is possible to detect framing errors (FER), parity errors (PER), rece
data full, receive data ready, and breaks.
- In transmitting, it is possible to detect transmit FIFO data empty.
- The number of stored data bytes is indicated for both the transmit and receive FIF
- An internal or external clock can be selected as the SCIF clock source.
  - When an internal clock is selected, the SCIF operates using the on-chip baud ra
  generator, and can output a serial clock signal with a frequency 16 times the bi
  - When an external clock is selected, the external clock input must have a freque
  the bit rate. (The on-chip baud rate generator is not used.)

Table 16.7 SCSMR Settings and SCIF Communication Formats

	SCSMR Settings					SCIF Communicatio
Mode	Bit 6 CHR	Bit 5 PE	Bit 3 STOP	Data Length	Parity Bit	Stop Bit Length
Asynchronous	0	0	0	8-bit	Not set	1 bit
			1			2 bits
		1	0		Set	1 bit
			1			2 bits
	1	0	0	7-bit	Not set	1 bit
			1			2 bits
		1	0		Set	1 bit
			1			2 bits

	1	bit rate

16.3.2 Serial Operation

Transmit/Receive Formats: Table 16.9 lists the eight communication formats that can
selected. The format is selected by settings in the serial mode register (SCSMR).

Table 16.9 Serial Communication Formats

SCSMR Bits			Serial Transmit/Receive Format and Frame Length										
CHR	PE	STOP	1	2	3	4	5	6	7	8	9	10	11
0	0	0	START	8-bit data								STOP	
0	0	1	START	8-bit data								STOP	STOP
0	1	0	START	8-bit data								P	STOP
0	1	1	START	8-bit data								P	STOP
1	0	0	START	7-bit data							STOP		
1	0	1	START	7-bit data							STOP	STOP	
1	1	0	START	7-bit data							P	STOP	
1	1	1	START	7-bit data							P	STOP	STOP

START: Start bit
STOP: Stop bit
P: Parity bit

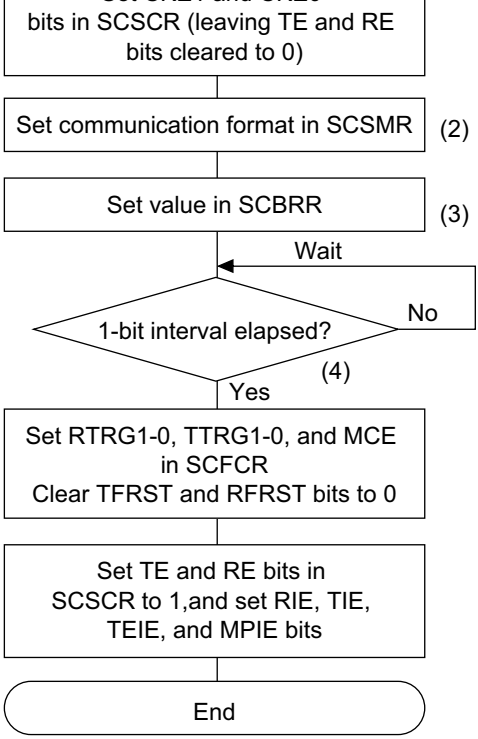
Transmitting and Receiving Data (SCIF Initialization): Before transmitting or rece
the TE and RE bits to 0 in the serial control register (SCSCR), then initialize the SCIF

When changing the communication format, always clear the TE and RE bits to 0 befo
the procedure given below. Clearing TE to 0 initializes the transmit shift register (SCT
Clearing TE and RE to 0, however, does not initialize the serial status register (SCSSR
FIFO data register (SCFTDR), or receive FIFO data register (SCFRDR), which retain
previous contents. Clear TE to 0 after all transmit data has been transmitted and the TI
the SCSSR is set. The TE bit can be cleared to 0 during transmission, but the transmit
the high impedance state after the bit is cleared to 0. Set the TFRST bit in SCFCR to 1
SCFTDR before TE is set again to start transmission.

When an external clock is used, the clock should not be stopped during initialization o
operation. SCIF operation becomes unreliable if the clock is stopped.

Figure 16.5 shows a sample flowchart for initializing the SCIF. The procedure for init
SCIF is:

1. Set the clock selection in SCSCR.
   Be sure to clear bits RIE TIE, TE, and RE to 0.
   When clock output is selected, the clock is output immediately after SCSCR settin
2. Set the communication format in SCSMR.
3. Write a value corresponding to the bit rate into the bit rate register (SCBRR).
   (Not necessary if an external clock is used.)
4. Wait at least one bit interval, then set the TE bit or RE bit in SCSCR to 1. Also se
   and TIE bits.
   Setting the TE and RE bits enables the TxD and RxD pins to be used. When transi
   SCIF will go to the mark state; when receiving, it will go to the idle state, waiting
   bit.



Note: Numbers in parentheses refer to steps in the preceding procedure description.

Figure 16.5 Sample Flowchart for SCIF Initialization

flags, then clear these flags to 0.

The number of transmit data bytes that can be written is (16 - transmit trigger set n

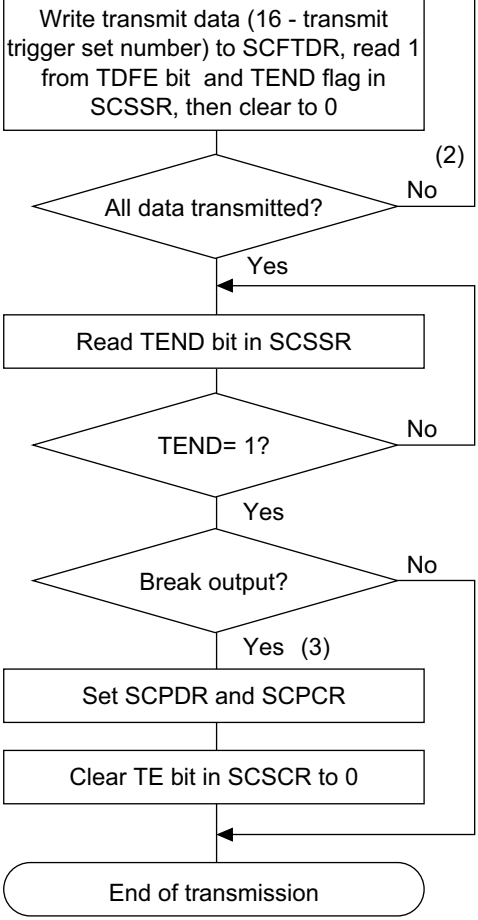
2. Serial transmission continuation procedure:

   To continue serial transmission, read 1 from the TDFE flag to confirm that writing then write data to SCFTDR, and then clear the TDFE flag to 0.

3. Break output at the end of serial transmission:

   To output a break in serial transmission, set the port SC data register (SCPDR) and control register (SCPCR), then clear the TE bit to 0 in the serial control register (S information on SCPDR and SCPCR, see section 16.2.8, Bit Rate Register (SCBRR

In steps 1 and 2, it is possible to ascertain the number of data bytes that can be written number of transmit data bytes in SCFTDR indicated by the upper 8 bits of the FIFO da register (SCFDR).



Note: Numbers in parentheses refer to steps in the preceding procedure description.

Figure 16.6 Sample Flowchart for Transmitting Serial Data

number of transmit data bytes in SCFTDR falls below the transmit trigger number
FIFO control register (SCFCR), the TDFE flag is set. If the TIE bit in the serial co
register (SCSR) is set to 1 at this time, a transmit-FIFO-data-empty interrupt (TXI
generated.

The serial transmit data is sent from the TxD pin in the following order.

a. Start bit: One-bit 0 is output.
b. Transmit data: 8-bit or 7-bit data is output in LSB-first order.
c. Parity bit: One parity bit (even or odd parity) is output. (A format in which a p
not output can also be selected.)
d. Stop bit(s): One or two 1-bits (stop bits) are output.
e. Mark state: 1 is output continuously until the start bit that starts the next transm
sent.

3. The SCIF checks the SCFTDR transmit data at the timing for sending the stop bit.
present, the data is transferred from SCFTDR to SCTSR, the stop bit is sent, and tl
transmission of the next frame is started.

   If there is no transmit data, the TEND flag in SCSSR is set to 1, the stop bit is sent
   the line goes to the mark state in which 1 is output continuously.

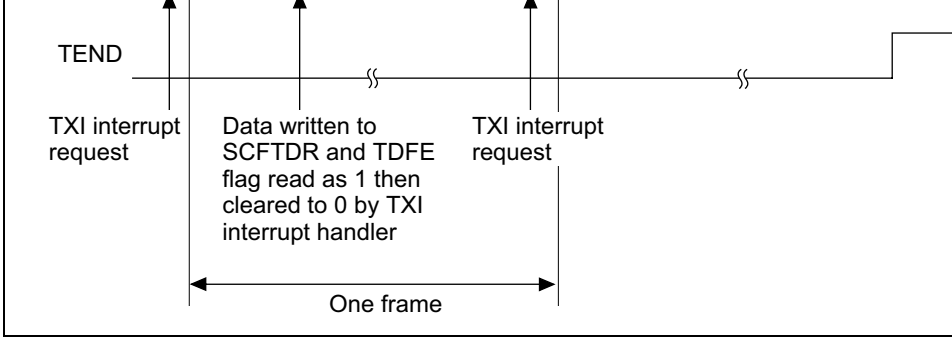


Figure 16.7 Example of Transmit Operation (8-Bit Data, Parity, One Stop Bit)

4. When modem control is enabled, transmission can be stopped and restarted in accor the $\overline{CTS}$ input value. When $\overline{CTS}$ is set to 1, if transmission is in progress, the line g mark state after transmission of one frame. When $\overline{CTS}$ is set to 0, the next transmit output starting from the start bit.

   Figure 16.8 shows an example of the operation when modem control is used.

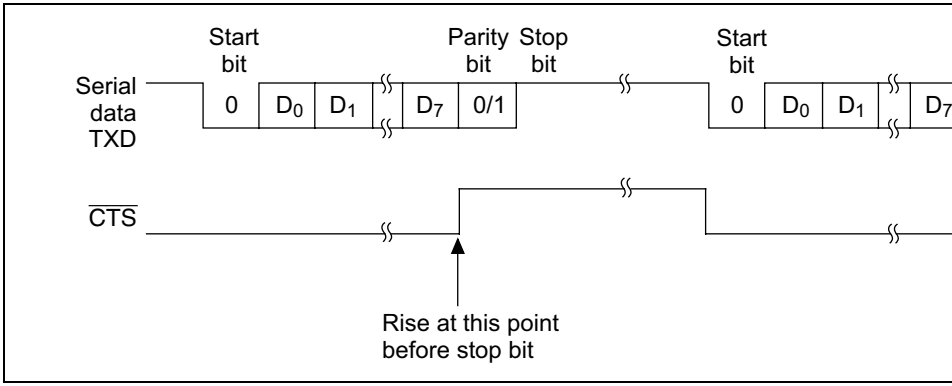
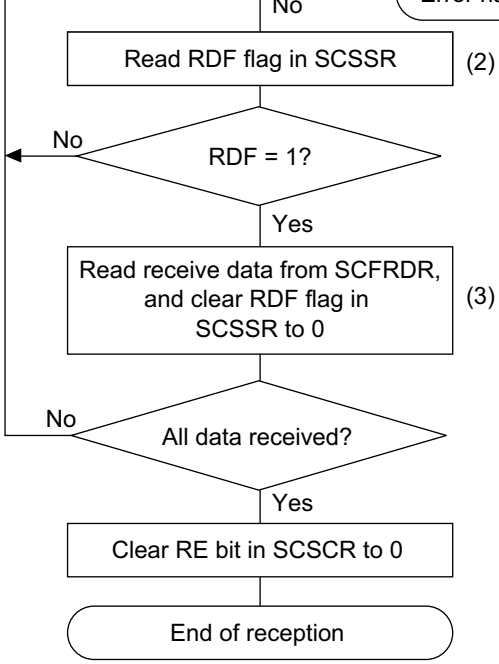


Figure 16.8 Example of Operation Using Modem Control ($\overline{CTS}$)

2. SCIF status check and receive data read : Read the serial status register (SCSSR) a that RDF = 1, then read the receive data in the receive FIFO data register (SCFRD from the RDF flag, and then clear the RDF flag to 0. The transition of the RDF fla 1 can be identified by an RXI interrupt.

3. Serial reception continuation procedure: To continue serial reception, read at least trigger set number of receive data bytes from SCFRDR, read 1 from the RDF flag, the RDF flag to 0. The number of receive data bytes in SCFRDR can be ascertain reading the lower bits of SCFDR.



Note: Numbers in parentheses refer to steps in the preceding procedure description.

Figure 16.9 Sample Flowchart for Receiving Serial Data

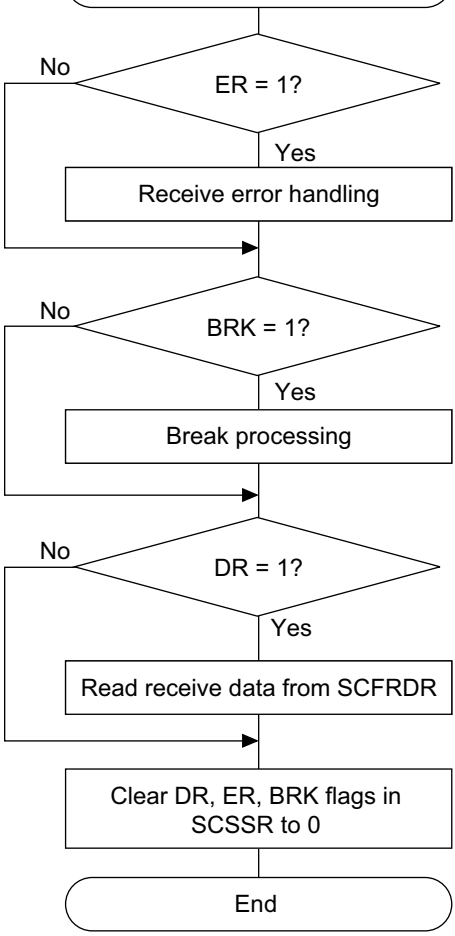


Figure 16.10 Sample Flowchart for Receiving Serial Data (cont)

a. Stop bit check: The SCIF checks whether the stop bit is 1. If there are two stop
the first is checked.
b. The SCIF checks whether receive data can be transferred from the receive shift
(SCRSR) to SCFRDR.
c. Break check: The SCIF checks that the BRK flag is 0, indicating that the break s
set.

If all the above checks are passed, the receive data is stored in SCFRDR.

Note: Reception is not suspended when a receive error occurs.

4. If the RIE bit in SCSR is set to 1 when the RDF or DR flag changes to 1, a receive-
full interrupt (RXI) request is generated.
If the RIE bit in SCSR is set to 1 when the ER flag changes to 1, a receive-error inte
request is generated.

If the RIE bit in SCSR is set to 1 when the BRK flag changes to 1, a break reception
(BRI) request is generated.

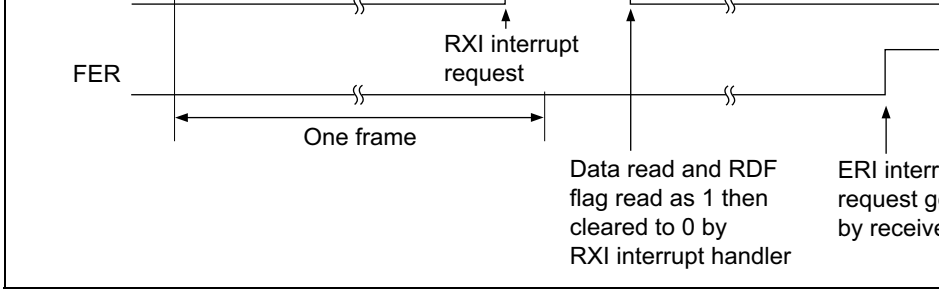


Figure 16.11 Example of SCIF Receive Operation (8-Bit Data, Parity, One Stop Bit)

5. When modem control is enabled, the $\overline{\text{RTS}}$ signal is output when SCFRDR is empty $\overline{\text{RTS}}$ is 0, reception is possible. When $\overline{\text{RTS}}$ is 1, this indicates that SCFRDR is full reception is not possible.

   Figure 16.12 shows an example of the operation when modem control is used.

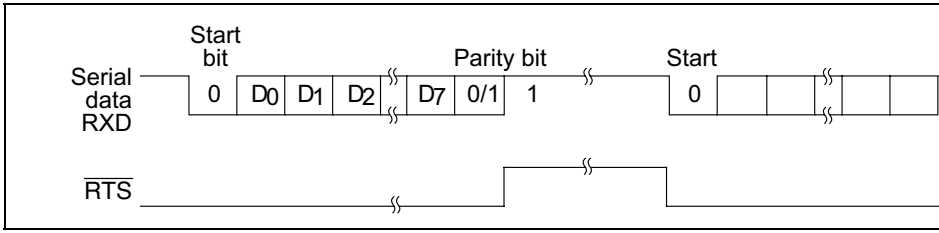


Figure 16.12 Example of Operation Using Modem Control ($\overline{\text{RTS}}$)

When the TDFE flag in the serial status register (SCSSR) is set to 1, a TXI interrupt re
generated. The DMAC can be activated and data transfer performed when this interrup
generated. When data exceeding the transmit trigger number is written to the transmit d
(SCFTDR) by the DMAC, 1 is read from the TDFE flag, after which 0 is written to it to

When the RDF flag in SCSSR is set to 1, an RXI interrupt request is generated. The DI
be activated and data transfer performed when the RDF flag in SCSSR is set to 1. When
data less than the receive trigger number is read from the receive data register (SCFRDR
DMAC, 1 is read from the RDF flag, after which 0 is written to it to clear it.

When the ER flag in SCSSR is set to 1, an ERI interrupt request is generated.

When the BRK flag in SCSSR is set to 1, a BRI interrupt request is generated.

The TXI interrupt indicates that transmit data can be written, and the RXI interrupt indi
there is receive data in SCFRDR.

Table 16.10 SCIF Interrupt Sources

Interrupt Source	Description	DMAC Activation	Priori Reset
ERI	Interrupt initiated by receive error flag (ER)	Not possible	High
RXI	Interrupt initiated by receive data FIFO full flag (RDF) or data ready flag (DR)	Possible (RDF only)	
BRI	Interrupt initiated by break flag (BRK)	Not possible	
TXI	Interrupt initiated by transmit FIFO data empty flag (TDFE)	Possible	Low

See section 4, Exception Handling, for priorities and the relationship to non-SCIF interr

However, if the number of data bytes written to SCFTDR is equal to or less than the t
trigger number, the TDFE flag will be set to 1 again even after having been cleared to
clearing should therefore be carried out after data exceeding the specified transmit trig
has been written to SCFTDR.
The number of transmit data bytes in SCFTDR can be found from the upper 8 bits of t
data count register (SCFDR).

2. SCFRDR Reading and RDF Flag: The RDF flag in the serial status register (SCS
when the number of receive data bytes in the receive FIFO data register (SCFRDR) ha
equal to or greater than the receive trigger number set by bits RTRG1 and RTRG0 in t
control register (SCFCR). After RDF is set, receive data equivalent to the trigger num
read from SCFRDR, allowing efficient continuous reception.
However, if the number of data bytes in SCFRDR exceeds the trigger number, the RD
be set to 1 again even after having been cleared to 0. RDF should therefore be cleared
being read as 1 after all the receive data has been read.
The number of receive data bytes in SCFRDR can be found from the lower 8 bits of th
count register (SCFDR).

3. Break Detection and Processing: Break signals can be detected by reading the Rx
directly when a framing error (FER) is detected. In the break state the input from the
consists of all 0s, so the FER flag is set and the parity error flag (PER) may also be set
although transfer of receive data to SCFRDR is halted in the break state, the SCIF rec
continues to operate, so if the BRK flag is cleared to 0 it will be set to 1 again.

4. Sending a Break Signal: The I/O condition and level of the TxD pin are determin
SCP4DT bit in the port SC data register (SCPDR) and bits SCP4MD0 and SCP4MD1
SC control register (SCPCR). This feature can be used to send a break signal.
To send a break signal during serial transmission, clear the SCP4DT bit to 0 (designat
level), then set the SCP4MD0 and SCP4MD1 bits to 0 and 1, respectively, and finally
bit to 0 (halting transmission). When the TE bit is cleared to 0, the transmitter is initia
regardless of the current transmission state, and 0 is output from the TxD pin.

edge of the eighth base clock pulse. The timing is shown in figure 16.13.

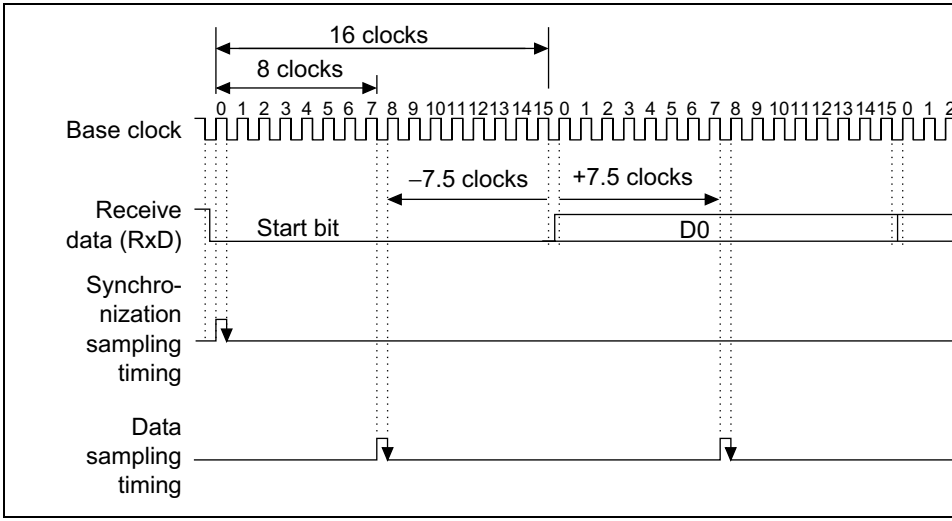


Figure 16.13 Receive Data Sampling Timing in Asynchronous Mode

The receive margin in asynchronous mode can therefore be expressed as shown in equa

Equation 1:

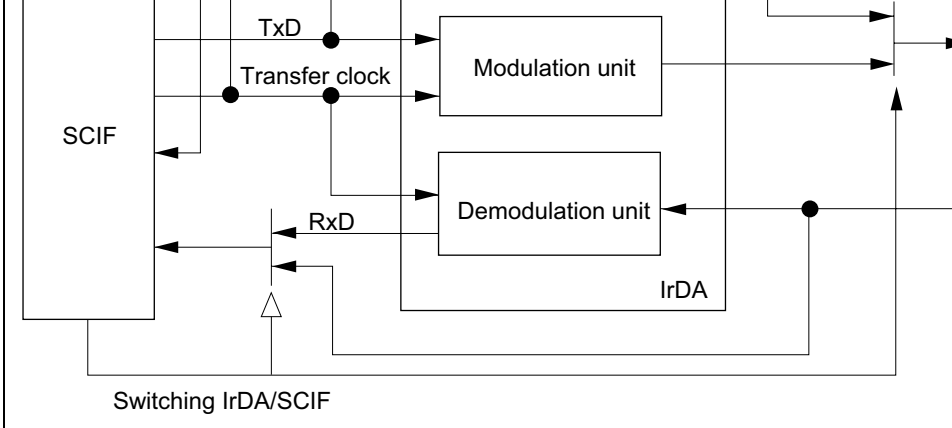
$$M = \left| \left(0.5 - \frac{1}{2N}\right) - (L - 0.5)\,F - \frac{|D - 0.5|}{N}(1 + F) \right| \times 100\%$$

Where: M: Receive margin (%)
N: Ratio of clock frequency to bit rate (N = 16)
D: Clock duty cycle (D = 0 to 1.0)
L: Frame length (L = 9 to 12)
F: Absolute deviation of clock frequency

From equation 1, if F = 0 and D = 0.5, the receive margin is 46.875%, as given by equa

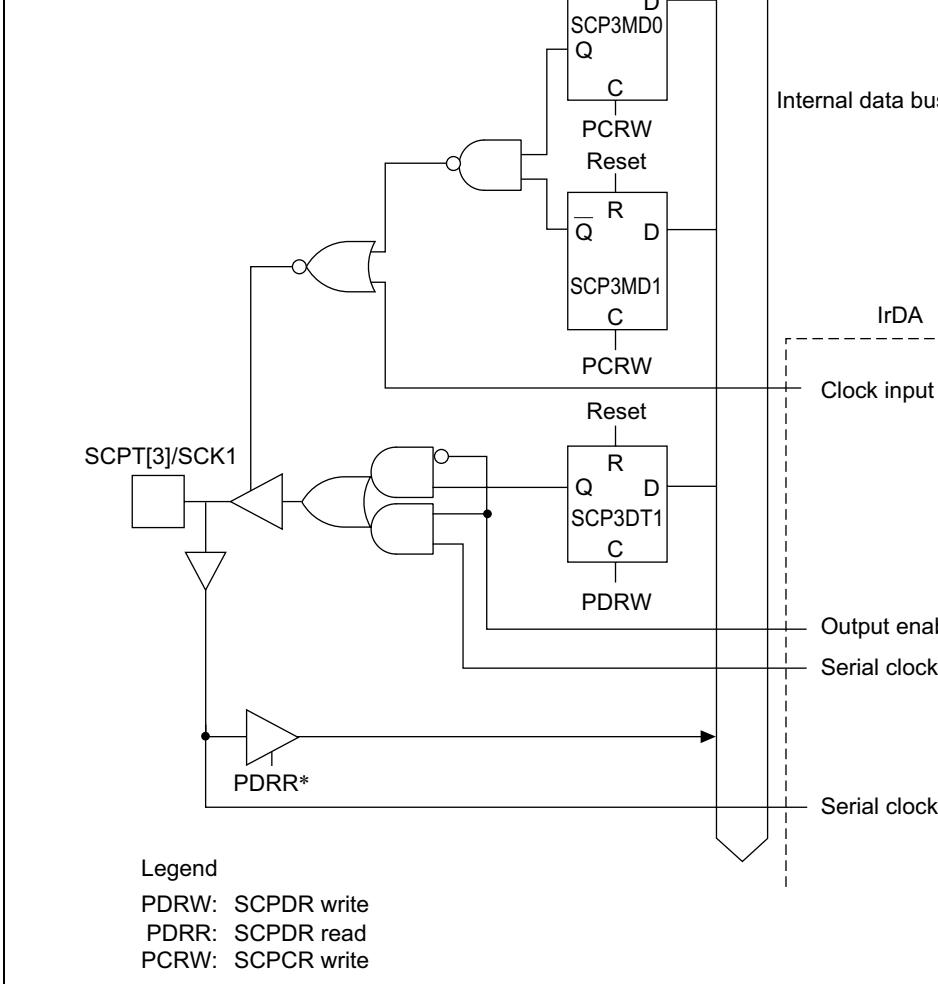
17.1.1 Features

- Conforms to the IrDA 1.0 system
- Asynchronous serial communication
  - Data length: 8 bits
  - Stop bit length: 1 bit
  - Parity bit: None
- On-chip 16-stage FIFO buffers for both transmit and receive operations
- On-chip baud rate generator with selectable bit rates
- Guard functions to protect the receiver during transmission
- Clock supply halted to reduce power consumption when not using the IrDA interfa



Legend
SCIF: Serial communication interface with FIFO

Figure 17.1 Block Diagram of IrDA



Note: * When reading the SCK1 pin, the CKE1 and CKE0 bits in SCSCR to 0, a
set the SCP3MD1 bit in SCPCR to 1 (see section 14.2.8, SC Port Contro
Register (SCPCR)/SC Port Data Register (SCPDR)).

Figure 17.2 SCPT[3]/SCK1 Pin

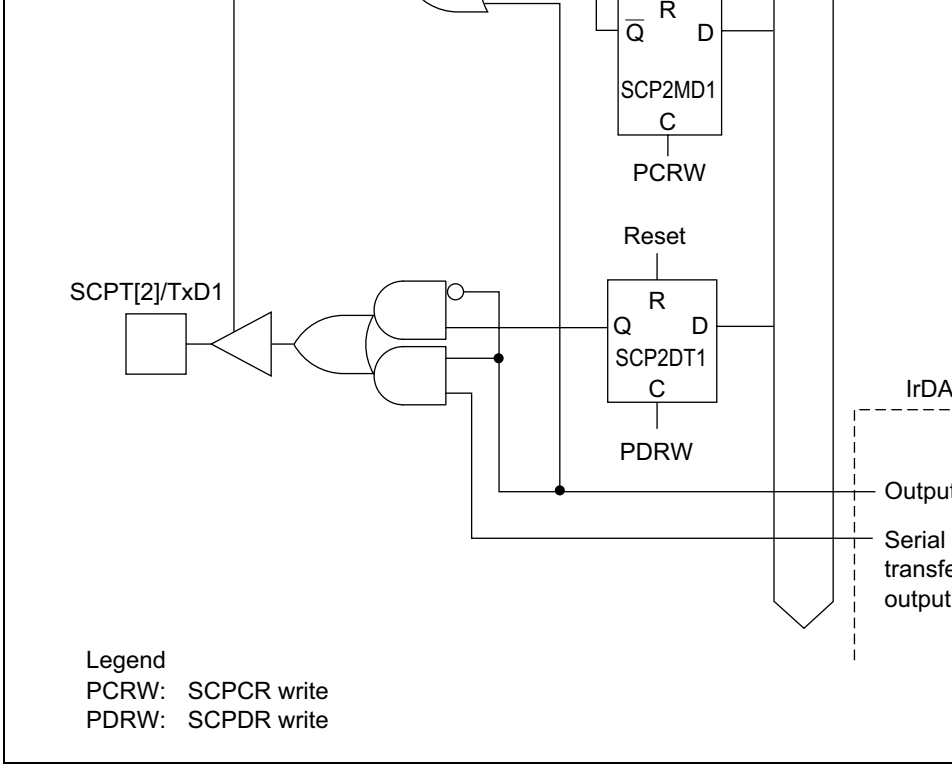
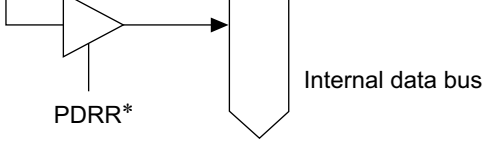


Figure 17.3 SCPT[2]/TxD1 Pin



Legend
PDRR: SCPDR read

Note: * When reading the RxD1 pin, set the RE bit in SCSCR to 1.

Figure 17.4 SCPT[2]/RxD1 Pin

17.1.3 Pin Configuration

The IrDA has the serial pins summarized in table 17.1.

Table 17.1 IrDA Pins

Pin Name	Signal Name	I/O	Function
Serial clock pin	SCK1	I/O	Clock I/O
Receive data pin	RxD1	Input	Receive data input
Transmit data pin	TxD1	Output	Transmit data output

Note: Clock input from the serial clock pin cannot be set in IrDA mode.

				(H'A4000140)[2]
Bit rate register 1	SCBRR1	R/W	H'FF	H'04000142 (H'A4000142)[2]
Serial control register 1	SCSCR1	R/W	H'00	H'04000144 (H'A4000144)[2]
Transmit FIFO data register 1	SCFTDR1	W	—	H'04000146 (H'A4000146)[2]
Serial status register 1	SCSSR1	R/(W)*1	H'0060	H'04000148 (H'A4000148)[2]
Receive FIFO data register 1	SCFRDR1	R	Undefined	H'0400014A (H'A400014A)[2]
FIFO control register 1	SCFCR1	R/W	H'00	H'0400014C (H'A400014C)
FIFO data count register 1	SCFDR1	R	H'0000	H'0400014E (H'A400014E)[2]

Notes: These registers are located in area 1 of physical space. Therefore, when the ca either access these registers from the P2 area of logical space or else make an setting using the MMU so that these registers are not cached.

1. Only 0 can be written to clear the flag.
2. When address translation by the MMU does not apply, the address in parent should be used.

Bit:	7	6	5	4	3	2	1
	IRMOD	ICK3	ICK2	ICK1	ICK0	PSEL	CKS
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

SCSMR is an 8-bit register that selects IrDA or SCIF mode, specifies the SCIF serial communication format, selects the IrDA output pulse width, and selects the baud rate clock source.

This module operates as IrDA when the IRMOD bit is set to 1. At this time, bits 3 to 0. This register functions in the same way as the SCSMR register in the SCIF when th bit is cleared to 0; therefore, this module can also operate as an SCIF.

SCSMR is initialized to H'00 by a power-on reset or manual reset, when the module is the module standby function, and in standby mode.

Bit 7—IrDA Mode (IRMOD): Selects whether this module operates as an IrDA seria communication interface or as an SCIF.

Bit 7: IRMOD	Description	
0	Operates as an SCIF	(
1*	Operates as an IrDA	

Note: * Do not set the CKE1 bit in the serial control register (SCSCRT) to 1 if the IRM to 1.

ICK3	ICK2	ICK1	ICK0	PSEL	Pulse width: 3/16 of 115 kbps bit leng
ICK3	ICK2	ICK1	ICK0	1	
Don't care	Don't care	Don't care	Don't care	0	Pulse width: 3/16 of bit length

It is necessary to generate a fixed clock pulse, IRCLK, by dividing the Pϕ clock by 1/2I the value of N determined by the setting of ICK3–ICK0).

Example:

Pϕ clock: 14.7456 MHz

IRCLK: 921.6 kHz (fixed)

N: Setting of ICK3–ICK0 ($0 \leq N \leq 15$)

$$N \geq \frac{P\phi}{2XIRCLK} - 1 \geq 7$$

Accordingly, N is 7.

Bits 1 and 0—Clock Select 1 and 0 (CKS1, CKS0): Select the internal baud rate gene source. Pϕ, Pϕ/4, Pϕ/16, or Pϕ/64 can be selected by setting the CKS1 and CKS0 bits.

Refer to section 14.2.9, Bit Rate Register (SCBRR), for the relationship between the clo the bit rate register set value, and the baud rate.

Bit 1: CKS1	Bit 0: CKS0	Description
0	0	Pϕ clock (In
0	1	Pϕ/4 clock
1	0	Pϕ/16
1	1	Pϕ/64

Note: Pϕ: Peripheral clock

Refer to section 16.3, Operation, for SCIF mode operation.

17.3.1 Overview

The IrDA module modifies TxD/RxD transmit/receive data waveforms to satisfy the I specification for infrared communication.

In the IrDA 1.0 specification, communication is first performed at a speed of 9600 bps communication speed is changed. However, the communication rate cannot be autom changed in this module, so the communication speed should be confirmed, and the ap speed set for this module by software.

Note: In IrDA mode, reception cannot be performed when the TE bit in the serial co (SCSCR) is set to 1 (enabling transmission). When performing reception, clea in SCSCR to 0.
As the SH7709S's RxD1 pin is active-high in IrDA mode, a (Schmidt) inverte inserted when connecting an active-low IrDA module.
The RxD1 pin is active-low in SCIF mode.

17.3.2 Transmitting

In the case of a serial output signal (UART frame) from the SCIF, its waveforms are n the signal is converted into the IR frame serial output signal by the IrDA module, as sh figure 17.5.

When serial data is 0, a pulse of 3/16 the IR frame bit width is generated and output. W data is 1, no pulse is output.

An infrared LED is driven by this signal demodulated to 3/16 width.

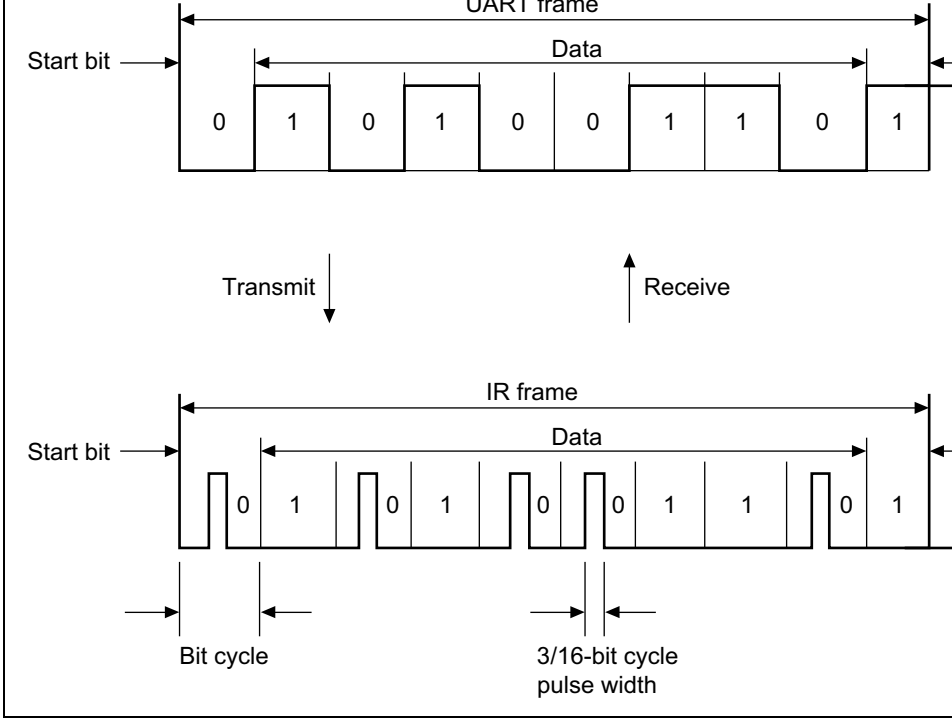


Figure 17.5 Transmit/Receive Operation

Table 18.1 List of Multiplexed Pins

Port	Port Function (Related Module)	Other Function (Related Module)
A	PTA7 input/output (port)	D23 input/output (data bus)
A	PTA6 input/output (port)	D22 input/output (data bus)
A	PTA5 input/output (port)	D21 input/output (data bus)
A	PTA4 input/output (port)	D20 input/output (data bus)
A	PTA3 input/output (port)	D19 input/output (data bus)
A	PTA2 input/output (port)	D18 input/output (data bus)
A	PTA1 input/output (port)	D17 input/output (data bus)
A	PTA0 input/output (port)	D16 input/output (data bus)
B	PTB7 input/output (port)	D31 input/output (data bus)
B	PTB6 input/output (port)	D30 input/output (data bus)
B	PTB5 input/output (port)	D29 input/output (data bus)
B	PTB4 input/output (port)	D28 input/output (data bus)
B	PTB3 input/output (port)	D27 input/output (data bus)
B	PTB2 input/output (port)	D26 input/output (data bus)
B	PTB1 input/output (port)	D25 input/output (data bus)
B	PTB0 input/output (port)	D24 input/output (data bus)
C	PTC7 input/output (port)/PINT7 input (INTC)	$\overline{\text{MCS7}}$ output (BSC)
C	PTC6 input/output (port)/PINT6 input (INTC)	$\overline{\text{MCS6}}$ output (BSC)
C	PTC5 input/output (port)/PINT5 input (INTC)	$\overline{\text{MCS5}}$ output (BSC)
C	PTC4 input/output (port)/PINT4 input (INTC)	$\overline{\text{MCS4}}$ output (BSC)
C	PTC3 input/output (port)/PINT3 input (INTC)	$\overline{\text{MCS3}}$ output (BSC)
C	PTC2 input/output (port)/PINT2 input (INTC)	$\overline{\text{MCS2}}$ output (BSC)
C	PTC1 input/output (port)/PINT1 input (INTC)	$\overline{\text{MCS1}}$ output (BSC)

D	PTD2 input/output (port)	$\overline{\text{RESETOUT}}$ output
D	PTD1 input/output (port)	DRAK0 output (DMAC)
D	PTD0 input/output (port)	DRAK1 output (DMAC)
E	PTE7 input/output (port)	$\overline{\text{AUDSYNC}}$ output (AUD)
E	PTE6 input/output (port)	—
E	PTE5 input/output (port)	$\overline{\text{CE2B}}$ output (PCMCIA)
E	PTE4 input/output (port)	$\overline{\text{CE2A}}$ output (PCMCIA)
E	PTE3 input/output (port)	—
E	PTE2 input/output (port)	$\overline{\text{RAS3U}}$ output (BSC)
E	PTE1 input/output (port)	—
E	PTE0 input/output (port)	TDO output (UDI)
F	PTF7 input (port)/PINT15 input (INTC)	$\overline{\text{TRST}}$ input (AUD, UDI)
F	PTF6 input (port)/PINT14 input (INTC)	TMS input (UDI)
F	PTF5 input (port)/PINT13 input (INTC)	TD1 input (UDI)
F	PTF4 input (port)/PINT12 input (INTC)	TCK input (UDI)
F	PTF3 input (port)/PINT11 input (INTC)	$\overline{\text{IRLS3}}$ input (INTC)
F	PTF2 input (port)/PINT10 input (INTC)	$\overline{\text{IRLS2}}$ input (INTC)
F	PTF1 input (port)/PINT9 input (INTC)	$\overline{\text{IRLS1}}$ input (INTC)
F	PTF0 input (port)/PINT8 input (INTC)	$\overline{\text{IRLS0}}$ input (INTC)
G	PTG7 input (port)	$\overline{\text{IOIS16}}$ input (PCMCIA)
G	PTG6 input (port)	$\overline{\text{ASEMD0}}$ input (AUD, UDI)
G	PTG5 input (port)	$\overline{\text{ASEBRKAK}}$ output (AUD)
G	PTG4 input (port)	CKIO2 output (CPG)
G	PTG3 input (port)	AUDATA3 output (AUD)
G	PTG2 input (port)	AUDATA2 output (AUD)

H	PTH3 input (port)/IRQ3 input (INTC)	IRQ3 input (INTC)
H	PTH2 input (port)/IRQ2 input (INTC)	IRQ2 input (INTC)
H	PTH1 input (port)/IRQ1 input (INTC)	IRQ1 input (INTC)
H	PTH0 input (port)/IRQ0 input (INTC)	IRQ0 input (INTC)
J	PTJ7 input/output (port)	STATUS1 output (CPG)
J	PTJ6 input/output (port)	STATUS0 output (CPG)
J	PTJ5 input/output (port)	—
J	PTJ4 input/output (port)	—
J	PTJ3 input/output (port)	$\overline{\text{CASU}}$ output (BSC)
J	PTJ2 input/output (port)	$\overline{\text{CASL}}$ output (BSC)
J	PTJ1 input/output (port)	—
J	PTJ0 input/output (port)	$\overline{\text{RAS3L}}$ output (BSC)
K	PTK7 input/output (port)	$\overline{\text{WE3}}$ output (BSC)/DQMUU outp (BSC)/$\overline{\text{ICIOWR}}$ output (BSC)
K	PTK6 input/output (port)	$\overline{\text{WE2}}$ output (BSC)/DQMUL outp (BSC)/$\overline{\text{ICIORD}}$ output (BSC)
K	PTK5 input/output (port)	CKE output (BSC)
K	PTK4 input/output (port)	$\overline{\text{BS}}$ output (BSC)
K	PTK3 input/output (port)	$\overline{\text{CS5}}$ output (BSC)/$\overline{\text{CE1A}}$ output
K	PTK2 input/output (port)	$\overline{\text{CS4}}$ output (BSC)
K	PTK1 input/output (port)	$\overline{\text{CS3}}$ output (BSC)
K	PTK0 input/output (port)	$\overline{\text{CS2}}$ output (BSC)
L	PTL7 input (port)	AN7 input (ADC)/DA0 output (D
L	PTL6 input (port)	AN6 input (ADC)/DA1 output (D
L	PTL5 input (port)	AN5 input (ADC)

SCPT	SCPT6 input/output (port)	$\overline{\text{RTS2}}$ output (UART ch 3)
SCPT	SCPT5 input/output (port)	SCK2 input/output (UART ch 3)
SCPT	SCPT4 input (port)	RxD2 input (UART ch 3)
	SCPT4 output (port)	TxD2 output (UART ch 3)
SCPT	SCPT3 input/output (port)	SCK1 input/output (UART ch 2)
SCPT	SCPT2 input (port)	RxD1 input (UART ch 2)
	SCPT2 output (port)	TxD1 output (UART ch 2)
SCPT	SCPT1 input/output (port)	SCK0 input/output (UART ch 1)
SCPT	SCPT0 input (port)	RxD0 input (UART ch 1)
	SCPT0 output (port)	TxD0 output (UART ch 1)

Note: SCPT0, SCPT2, and SCPT4 have the same data register to be accessed althou
have different input pins and output pins.

				(H'A4000100)
Port B control register	PBCR	R/W	H'0000	H'04000102 (H'A4000102)*
Port C control register	PCCR	R/W	H'AAAA	H'04000104 (H'A4000104)*
Port D control register	PDCR	R/W	H'AA8A	H'04000106 (H'A4000106)*
Port E control register	PECR	R/W	H'AAAA/H'2AA8	H'04000108 (H'A4000108)*
Port F control register	PFCR	R/W	H'AAAA/H'00AA	H'0400010A (H'A400010A)*
Port G control register	PGCR	R/W	H'AAAA/H'A200	H'0400010C (H'A400010C)*
Port H control register	PHCR	R/W	H'AAAA/H'8AAA	H'0400010E (H'A400010E)*
Port J control register	PJCR	R/W	H'0000	H'04000110 (H'A4000110)*
Port K control register	PKCR	R/W	H'0000	H'04000112 (H'A4000112)*
Port L control register	PLCR	R/W	H'0000	H'04000114 (H'A4000114)*
SC port control register	SCPCR	R/W	H'A888	H'04000116 (H'A4000116)*

Notes: 1. The initial value of the port E, F, G, and H control registers depends on the $\overline{\text{ASEMD0}}$ pin.
If a low level is input at the $\overline{\text{ASEMD0}}$ pin while the $\overline{\text{RESETP}}$ pin is asserted, is entered; if a high level is input, normal mode is entered. See section 22, Debugging Interface (UDI), for more information on the UDI.

2. These registers are located in area 1 of physical space. Therefore, when th on, either access these registers from the P2 area of logical space or else appropriate setting using the MMU so that these registers are not cached.

* When address translation by the MMU does not apply, the address in parer should be used.

R/W: R/W R/W R/W R/W R/W R/W R/W R/W R/W R/W R/W R/W R/W R/W R/W

The port A control register (PACR) is a 16-bit readable/writable register that selects the functions. PACR is initialized to H'0000 by a power-on reset, but is not initialized by a reset, in standby mode, or in sleep mode.

Bits 15 and 14—PA7 Mode 1 and 0 (PA7MD1, PA7MD0)
Bits 13 and 12—PA6 Mode 1 and 0 (PA6MD1, PA6MD0)
Bits 11 and 10—PA5 Mode 1 and 0 (PA5MD1, PA5MD0)
Bits 9 and 8—PA4 Mode 1 and 0 (PA4MD1, PA4MD0)
Bits 7 and 6—PA3 Mode 1 and 0 (PA3MD1, PA3MD0)
Bits 5 and 4—PA2 Mode 1 and 0 (PA2MD1, PA2MD0)
Bits 3 and 2—PA1 Mode 1 and 0 (PA1MD1, PA1MD0)
Bits 1 and 0—PA0 Mode 1 and 0 (PA0MD1, PA0MD0)
These bits select the pin functions and perform input pull-up MOS control.

Bit (2n + 1)	Bit 2n		
PAnMD1	**PAnMD0**	**Pin Function**	
0	0	Other function (see table 18.1)	(In
0	1	Port output	
1	0	Port input (Pull-up MOS: on)	
1	1	Port input (Pull-up MOS: off)	

The port B control register (PBCR) is a 16-bit readable/writable register that selects th
functions. PBCR is initialized to H'0000 by a power-on reset, but is not initialized by a
reset, in standby mode, or in sleep mode.

Bits 15 and 14—PB7 Mode 1 and 0 (PB7MD1, PB7MD0)
Bits 13 and 12—PB6 Mode 1 and 0 (PB6MD1, PB6MD0)
Bits 11 and 10—PB5 Mode 1 and 0 (PB5MD1, PB5MD0)
Bits 9 and 8—PB4 Mode 1 and 0 (PB4MD1, PB4MD0)
Bits 7 and 6—PB3 Mode 1 and 0 (PB3MD1, PB3MD0)
Bits 5 and 4—PB2 Mode 1 and 0 (PB2MD1, PB2MD0)
Bits 3 and 2—PB1 Mode 1 and 0 (PB1MD1, PB1MD0)
Bits 1 and 0—PB0 Mode 1 and 0 (PB0MD1, PB0MD0)
These bits select the pin functions and perform input pull-up MOS control.

Bit (2n + 1)	Bit 2n	
PBnMD1	**PBnMD0**	**Pin Function**
0	0	Other function (see table 18.1)
0	1	Port output
1	0	Port input (Pull-up MOS: on)
1	1	Port input (Pull-up MOS: off)

The port C control register (PCCR) is a 16-bit readable/writable register that selects the functions. PCCR is initialized to H'AAAA by a power-on reset, but is not initialized by reset, in standby mode, or in sleep mode.

Bits 15 and 14—PC7 Mode 1 and 0 (PC7MD1, PC7MD0)
Bits 13 and 12—PB6 Mode 1 and 0 (PC6MD1, PC6MD0)
Bits 11 and 10—PC5 Mode 1 and 0 (PC5MD1, PC5MD0)
Bits 9 and 8—PC4 Mode 1 and 0 (PC4MD1, PC4MD0)
Bits 7 and 6—PC3 Mode 1 and 0 (PC3MD1, PC3MD0)
Bits 5 and 4—PC2 Mode 1 and 0 (PC2MD1, PC2MD0)
Bits 3 and 2—PC1 Mode 1 and 0 (PC1MD1, PC1MD0)
Bits 1 and 0—PC0 Mode 1 and 0 (PC0MD1, PC0MD0)
These bits select the pin functions and perform input pull-up MOS control.

Bit (2n + 1)	Bit 2n	
PCnMD1	**PCnMD0**	**Pin Function**
0	0	Other function (see table 18.1)
0	1	Port output
1	0	Port input (Pull-up MOS: on) (In
1	1	Port input (Pull-up MOS: off)

The port D control register (PDCR) is a 16-bit readable/writable register that selects th
functions. PDCR is initialized to H'AA8A by a power-on reset, but is not initialized by
reset, in standby mode, or in sleep mode.

Bits 15 and 14—PD7 Mode 1 and 0 (PD7MD1, PD7MD0)
Bits 11 and 10—PD5 Mode 1 and 0 (PD5MD1, PD5MD0)
Bits 7 and 6—PD3 Mode 1 and 0 (PD3MD1, PD3MD0)
Bits 5 and 4—PD2 Mode 1 and 0 (PD2MD1, PD2MD0)
Bits 3 and 2—PD1 Mode 1 and 0 (PD1MD1, PD1MD0)
Bits 1 and 0—PD0 Mode 1 and 0 (PD0MD1, PD0MD0)
These bits select the pin functions and perform input pull-up MOS control.

Bit (2n + 1)	Bit 2n		
PDnMD1	**PDnMD0**	**Pin Function**	
0	0	Other function (see table 18.1)	(Initial va
0	1	Port output	
1	0	Port input (Pull-up MOS: on)	(Initial value) (n =
1	1	Port input (Pull-up MOS: off)	

(n =

Bits 13 and 12—PD6 Mode 1 and 0 (PD6MD1, PD6MD0)
Bits 9 and 8—PD4 Mode 1 and 0 (PD4MD1, PD4MD0)
These bits select the pin functions and perform input pull-up MOS control.

Bit (2n + 1)	Bit 2n		
PDnMD1	**PDnMD0**	**Pin Function**	
0	0	Other function (see table 18.1)	
0	1	Reserved	
1	0	Port input (Pull-up MOS: on)	(
1	1	Port input (Pull-up MOS: off)	

The port E control register (PECR) is a 16-bit readable/writable register that selects the functions. PECR is initialized to H'AAAA ($\overline{\text{ASEMD0}}$ = 1) or H'2AA8 ($\overline{\text{ASEMD0}}$ = 0) power-on reset, but is not initialized by a manual reset, in software standby mode, or in mode.

Bits 15 and 14—PE7 Mode 1 and 0 (PE7MD1, PE7MD0)
Bits 13 and 12—PE6 Mode 1 and 0 (PE6MD1, PE6MD0)
Bits 11 and 10—PE5 Mode 1 and 0 (PE5MD1, PE5MD0)
Bits 9 and 8—PE4 Mode 1 and 0 (PE4MD1, PE4MD0)
Bits 7 and 6—PE3 Mode 1 and 0 (PE3MD1, PE3MD0)
Bits 5 and 4—PE2 Mode 1 and 0 (PE2MD1, PE2MD0)
Bits 3 and 2—PE1 Mode 1 and 0 (PE1MD1, PE1MD0)
Bits 1 and 0—PE0 Mode 1 and 0 (PE0MD1, PE0MD0)
These bits select the pin functions and perform input pull-up MOS control.

Bit (2n + 1)	Bit 2n		
PEnMD1	**PEnMD0**	**Pin Function**	
0	0	Reserved (n = 0, 7) (see table 18.1)	(Initial value) ($\overline{\text{ASI}}$
0	1	Port output	
1	0	Port input (Pull-up MOS: on)	(Initial value) ($\overline{\text{ASI}}$
1	1	Port input (Pull-up MOS: off)	

Bit (2n + 1)	Bit 2n		
PEnMD1	**PEnMD0**	**Pin Function**	
0	0	Other function (n = 2, 4, 5) (see table 18.1), Reserved (n =	
0	1	Port output	
1	0	Port input (Pull-up MOS: on)	(In
1	1	Port input (Pull-up MOS: off)	

The port F control register (PFCR) is a 16-bit readable/writable register that selects the functions. PFCR is initialized to H'AAAA (ASEMD0 = 1) or H'00AA (ASEMD0 = 0) on reset, but is not initialized by a manual reset, in standby mode, or in sleep mode.

Bits 15 and 14—PF7 Mode 1 and 0 (PF7MD1, PF7MD0)
Bits 13 and 12—PF6 Mode 1 and 0 (PF6MD1, PF6MD0)
Bits 11 and 10—PF5 Mode 1 and 0 (PF5MD1, PF5MD0)
Bits 9 and 8—PF4 Mode 1 and 0 (PF4MD1, PF4MD0)
Bits 7 and 6—PF3 Mode 1 and 0 (PF3MD1, PF3MD0)
Bits 5 and 4—PF2 Mode 1 and 0 (PF2MD1, PF2MD0)
Bits 3 and 2—PF1 Mode 1 and 0 (PF1MD1, PF1MD0)
Bits 1 and 0—PF0 Mode 1 and 0 (PF0MD1, PF0MD0)
These bits select the pin functions and perform input pull-up MOS control.

Bit (2n + 1)	Bit 2n		
PFnMD1	**PFnMD0**	**Pin Function**	
0	0	Other function (see table 18.1)	(Initial value) ($\overline{AS}$
0	1	Reserved	
1	0	Port input (Pull-up MOS: on)	(Initial value) ($\overline{AS}$
1	1	Port input (Pull-up MOS: off)	

Bit (2n + 1)	Bit 2n		
PFnMD1	**PFnMD0**	**Pin Function**	
0	0	Other function (see table 18.1)	
0	1	Reserved	
1	0	Port input (Pull-up MOS: on)	(
1	1	Port input (Pull-up MOS: off)	

The port G control register (PGCR) is a 16-bit readable/writable register that selects the functions. PGCR is initialized to H'AAAA ($\overline{\text{ASEMD0}}$ = 1) or H'A200 ($\overline{\text{ASEMD0}}$ = 0) on reset, but is not initialized by a manual reset, in standby mode, or in sleep mode.

Bits 15 and 14—PG7 Mode 1 and 0 (PG7MD1, PG7MD0)
Bits 13 and 12—PG6 Mode 1 and 0 (PG6MD1, PG6MD0)
Bits 11 and 10—PG5 Mode 1 and 0 (PG5MD1, PG5MD0)
Bits 9 and 8—PG4 Mode 1 and 0 (PG4MD1, PG4MD0)
Bits 7 and 6—PG3 Mode 1 and 0 (PG3MD1, PG3MD0)
Bits 5 and 4—PG2 Mode 1 and 0 (PG2MD1, PG2MD0)
Bits 3 and 2—PG1 Mode 1 and 0 (PG1MD1, PG1MD0)
Bits 1 and 0—PG0 Mode 1 and 0 (PG0MD1, PG0MD0)
These bits select the pin functions and perform input pull-up MOS control.

Bit (2n + 1)	Bit 2n	
PGnMD1	**PGnMD0**	**Pin Function**
0	0	Other function (n = 4, 6, 7) (see table 18.1)
0	1	Reserved
1	0	Port input (Pull-up MOS: on) (In
1	1	Port input (Pull-up MOS: off)

Note: * When n = 6, $\overline{\text{ASEMD0}}$/PTG6 functions as $\overline{\text{ASEMD0}}$ input while the reset signa asserted, and as PTG6 input immediately after the reset signal is nagated.

Bit 3	Bit 0	
PG1MD1*	**PG0MD0**	**Pin Function**
0	0	Other function (see table 18.1) (Initial value) $\overline{\text{A}}$
0	1	Reserved
1	0	Port input (Pull-up MOS: on) (Initial value) $\overline{\text{A}}$
1	1	Port input (Pull-up MOS: off)

Note: * Controlled by PG1MD1 (bit 3), not PG0MD1 (bit 1).

18.3.8 Port H Control Register (PHCR)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2
	PH7 MD1	PH7 MD0	PH6 MD1	PH6 MD0	PH5 MD1	PH5 MD0	PH4 MD1	PH4 MD0	PH3 MD1	PH3 MD0	PH2 MD1	PH2 MD0	PH1 MD1	PH MD
Initial value:	1	0	1/0	0	1	0	1	0	1	0	1	0	1	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/

The port H control register (PHCR) is a 16-bit readable/writable register that selects th functions. PHCR is initialized to H'AAAA ($\overline{\text{ASEMD0}}$ = 1) or H'8AAA ($\overline{\text{ASEMD0}}$ = 0 power-on reset, but is not initialized by a manual reset, in standby mode, or in sleep m

1	1	Port input (Pull-up MOS: off)

Bits 13 and 12—PH6 Mode 1 and 0 (PH6MD1, PH6MD0)
Bits 11 and 10—PH5 Mode 1 and 0 (PH5MD1, PH5MD0)
Bits 9 and 8—PH4 Mode 1 and 0 (PH4MD1, PH4MD0)
Bits 7 and 6—PH3 Mode 1 and 0 (PH3MD1, PH3MD0)
Bits 5 and 4—PH2 Mode 1 and 0 (PH2MD1, PH2MD0)
Bits 3 and 2—PH1 Mode 1 and 0 (PH1MD1, PH1MD0)
Bits 1 and 0—PH0 Mode 1 and 0 (PH0MD1, PH0MD0)
These bits select the pin functions and perform input pull-up MOS control.

Bit 13	Bit 12		
PH6MD1	**PH6MD0**	**Pin Function**	
0	0	Other function (see table 18.1)	(Initial value) ($\overline{\text{ASI}}$
0	1	Reserved	
1	0	Port input (Pull-up MOS: on)	(Initial value) ($\overline{\text{ASI}}$
1	1	Port input (Pull-up MOS: off)	

Bit (2n + 1)	Bit 2n		
PHnMD1	**PHnMD0**	**Pin Function**	
0	0	Other function (see table 18.1)	
0	1	Reserved	
1	0	Port input (Pull-up MOS: on)	(In
1	1	Port input (Pull-up MOS: off)	

The port J control register (PJCR) is a 16-bit readable/writable register that selects the functions. PJCR is initialized to H'0000 by a power-on reset, but is not initialized by a reset, in standby mode, or in sleep mode.

Bits 15 and 14—PJ7 Mode 1 and 0 (PJ7MD1, PJ7MD0)
Bits 13 and 12—PJ6 Mode 1 and 0 (PJ6MD1, PJ6MD0)
Bits 11 and 10—PJ5 Mode 1 and 0 (PJ5MD1, PJ5MD0)
Bits 9 and 8—PJ4 Mode 1 and 0 (PJ4MD1, PJ4MD0)
Bits 7 and 6—PJ3 Mode 1 and 0 (PJ3MD1, PJ3MD0)
Bits 5 and 4—PJ2 Mode 1 and 0 (PJ2MD1, PJ2MD0)
Bits 3 and 2—PJ1 Mode 1 and 0 (PJ1MD1, PJ1MD0)
Bits 1 and 0—PJ0 Mode 1 and 0 (PJ0MD1, PJ0MD0)
These bits select the pin functions and perform input pull-up MOS control.

Bit (2n + 1)	Bit 2n	
PJnMD1	**PJnMD0**	**Pin Function**
0	0	Other function (n = 0, 2, 3, 6, 7) (see table 18.1), Reserved (n = 1, 4, 5) (I
0	1	Port output
1	0	Port input (Pull-up MOS: on)
1	1	Port input (Pull-up MOS: off)

The port K control register (PKCR) is a 16-bit readable/writable register that selects the functions. PKCR is initialized to H'0000 by a power-on reset, but is not initialized by a reset, in standby mode, or in sleep mode.

Bits 15 and 14—PK7 Mode 1 and 0 (PK7MD1, PK7MD0)
Bits 13 and 12—PK6 Mode 1 and 0 (PK6MD1, PK6MD0)
Bits 11 and 10—PK5 Mode 1 and 0 (PK5MD1, PK5MD0)
Bits 9 and 8—PK4 Mode 1 and 0 (PK4MD1, PK4MD0)
Bits 7 and 6—PK3 Mode 1 and 0 (PK3MD1, PK3MD0)
Bits 5 and 4—PK2 Mode 1 and 0 (PK2MD1, PK2MD0)
Bits 3 and 2—PK1 Mode 1 and 0 (PK1MD1, PK1MD0)
Bits 1 and 0—PK0 Mode 1 and 0 (PK0MD1, PK0MD0)
These bits select the pin functions and perform input pull-up MOS control.

Bit (2n + 1)	Bit 2n	
PKnMD1	**PKnMD0**	**Pin Function**
0	0	Other function (see table 18.1) (In
0	1	Port output
1	0	Port input (Pull-up MOS: on)
1	1	Port input (Pull-up MOS: off)

The port L control register (PLCR) is a 16-bit readable/writable register that selects th functions. PLCR is initialized to H'0000 by a power-on reset, but is not initialized by reset, in standby mode, or in sleep mode.

Bits 15 and 14—PL7 Mode 1 and 0 (PL7MD1, PL7MD0)
Bits 13 and 12—PL6 Mode 1 and 0 (PL6MD1, PL6MD0)
Bits 11 and 10—PL5 Mode 1 and 0 (PL5MD1, PL5MD0)
Bits 9 and 8—PL4 Mode 1 and 0 (PL4MD1, PL4MD0)
Bits 7 and 6—PL3 Mode 1 and 0 (PL3MD1, PL3MD0)
Bits 5 and 4—PL2 Mode 1 and 0 (PL2MD1, PL2MD0)
Bits 3 and 2—PL1 Mode 1 and 0 (PL1MD1, PL1MD0)
Bits 1 and 0—PL0 Mode 1 and 0 (PL0MD1, PL0MD0)
These bits select the pin functions and perform input pull-up MOS control.

Bit (2n + 1)	Bit 2n	
PLnMD1	**PLnMD0**	**Pin Function**
0	0	Other function (see table 18.1) (I
0	1	Reserved
1	0	Port input (Pull-up MOS: on)
1	1	Port input (Pull-up MOS: off)

When the DA0 and DA1 pins are used as the D/A converter outputs or when PTL7 an used in the “other function” state, PLCR should by kept at its initial value.

The SC port control register (SCPCR) is a 16-bit readable/writable register that selects
functions. The setting of SCPCR is valid only when transmit/receive operations are disa
SCSCR register. SCPCR is initialized to H'A888 by a power-on reset, but is not initiali
manual reset, in standby mode, or in sleep mode. When the TE bit in SCSCR is set to 1
function" output state has a higher priority than the SCPCR setting for the TxD[2:0] pi
the RE bit in SCSCR is set to 1, the input state has a higher priority than the SCPCR se
RxD[2:0] pins.

Bits 15 and 14—SCP7 Mode 1 and 0 (SCP7MD1, SCP7MD0): These bits select the
function and perform input pull-up MOS control.

Bit 15	Bit 14		
SCP7MD1	**SCP7MD0**	**Pin Function**	
0	0	Other function (see table 18.1)	
0	1	Reserved	
1	0	Port input (Pull-up MOS: on)	(In
1	1	Port input (Pull-up MOS: off)	

Bits 13, 12—SCP6 Mode 1, 0 (SCP6MD1, SCP6MD0): These bits select the pin func
perform input pull-up MOS control.

Bit 13	Bit 12		
SCP6MD1	**SCP6MD0**	**Pin Function**	
0	0	Other function (see table 18.1)	
0	1	Port output	
1	0	Port input (Pull-up MOS: on)	(In
1	1	Port input (Pull-up MOS: off)	

1	1	Port input (Pull-up MOS: off)

Bits 9 and 8—SCP4 Mode 1 and 0 (SCP4MD1, SCP4MD0): These bits select the p
and perform input pull-up MOS control.

Bit 9	Bit 8	
SCP4MD1	**SCP4MD0**	**Pin Function**
0	0	Transmit data output 2 (TxD2) Receive data input 2 (RxD2)
0	1	General output (SCPT[4] output pin) Receive data input 2 (RxD2)
1	0	SCPT[4] input pin pull-up (input pin) Transmit data output 2 (TxD2)
1	1	General input (SCPT[4] input pin) Transmit data output 2 (TxD2)

Note: There is no SCPT[4] simultaneous I/O combination because one bit (SCP4DT)
using two pins, TxD2 and RxD2.

When port input is set (bit SCPnMD1 is set to 1) and when the TE bit in SCSCR is set
TxD2 pin is in the output state. When the TE bit is cleared to 0, the TxD2 pin goes to
impedance state.

Bits 7 and 6—SCP3 Mode 1 and 0 (SCP3MD1, SCP3MD0): These bits select the p
and perform input pull-up MOS control.

Bit 7	Bit 6	
SCP3MD1	**SCP3MD0**	**Pin Function**
0	0	Other function (see table 18.1)
0	1	Port output
1	0	Port input (Pull-up MOS: on)
1	1	Port input (Pull-up MOS: off)

1	0	SCPT[2] input pin pull-up (input pin) Transmit data output 1 (TxD1)
1	1	General input (SCPT[2] input pin) Transmit data output 1 (TxD1)

Note: There is no SCPT[2] simultaneous I/O combination because one bit (SCP2DT) is using two pins, TxD1 and RxD1.

When port input is set (bit SCPnMD1 is set to 1) and when the TE bit in SCSCR is set t TxD1 pin is in the output state. When the TE bit is cleared to 0, the TxD1 pin goes to t impedance state.

Bits 3 and 2—SCP1 Mode 1 and 0 (SCP1MD1, SCP1MD0): These bits select the pin and perform input pull-up MOS control.

Bit 3	Bit 2	
SCP1MD1	**SCP1MD0**	**Pin Function**
0	0	Other function (see table 18.1)
0	1	Port output
1	0	Port input (Pull-up MOS: on) (In
1	1	Port input (Pull-up MOS: off)

1	0	SCPT[0] input pin pull-up (input pin) Transmit data output 0 (TxD0)
1	1	General input (SCPT[0] input pin) Transmit data output 0 (TxD0)

Note: There is no SCPT[0] simultaneous I/O combination because one bit (SCP0DT) using two pins, TxD0 and RxD0.

When port input is set (bit SCPnMD1 is set to 1) and when the TE bit in SCSCR is set TxD0 pin is in the output state. When the TE bit is cleared to 0, the TxD0 pin goes to impedance state.

19.2 Port A

Port A is an 8-bit input/output port with the pin configuration shown in figure 19.1. Ea
an input pull-up MOS, which is controlled by the port A control register (PACR) in th

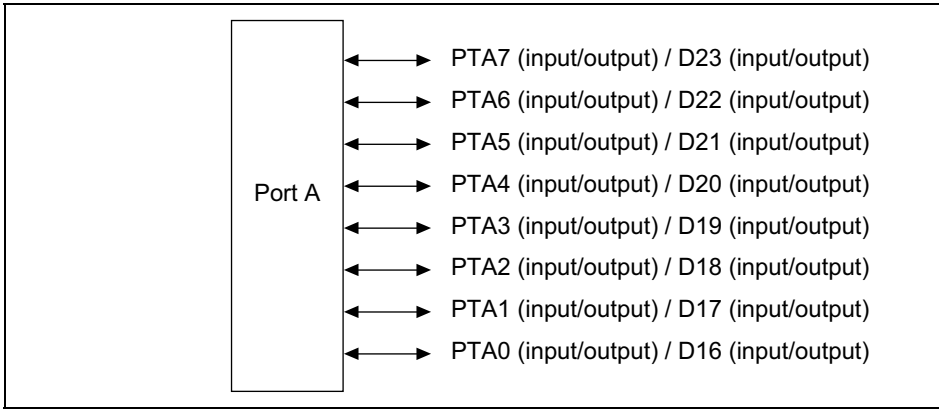


Figure 19.1 Port A

19.2.1 Register Description

Table 19.1 summarizes the port A register.

Table 19.1 Port A Register

Name	Abbreviation	R/W	Initial Value	Address	A
Port A data register	PADR	R/W	H'00	H'04000120 (H'A4000120)*	8

Notes: This register is located in area 1 of physical space. Therefore, when the cache
access this register from the P2 area of logical space or else make an appropr
using the MMU so that this register is not cached.

* When address translation by the MMU does not apply, the address in parent
should be used.

PTA7 to PTA0. Bits PA7DT to PA0DT correspond to pins PTA7 to PTA0. When the p
is general output port, if the port is read the value of the corresponding PADR bit is retu
directly. When the function is general input port, if the port is read the corresponding pi
read. Table 19.2 shows the function of PADR.

PADR is initialized to H'00 by a power-on reset. It retains its previous value in standby
sleep mode, and in a manual reset.

Table 19.2 Port A Data Register (PADR) Read/Write Operations

PAnMD1	PAnMD0	Pin State	Read	Write
0	0	Other function (See table 18.1)	PADR value	Value is written to PADR, bu affect pin state
	1	Output	PADR value	Write value is output from pi
1	0	Input (Pull-up MOS on)	Pin state	Value is written to PADR, bu affect pin state
	1	Input (Pull-up MOS off)	Pin state	Value is written to PADR, bu affect pin state

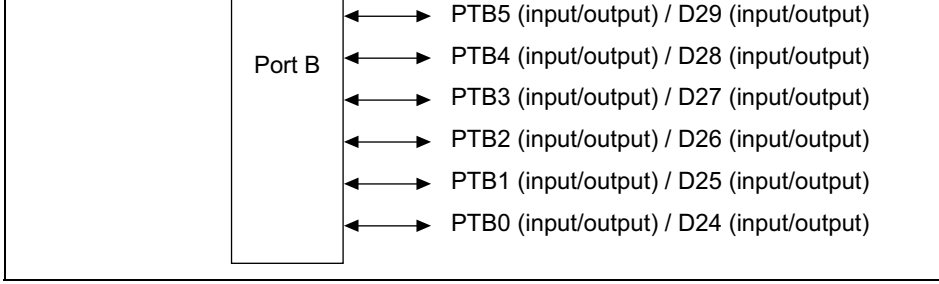


Figure 19.2 Port B

19.3.1 Register Description

Table 19.3 summarizes the port B register.

Table 19.3 Port B Register

Name	Abbreviation	R/W	Initial Value	Address	A
Port B data register	PBDR	R/W	H'00	H'04000122 (H'A4000122)*	8

Notes: This register is located in area 1 of physical space. Therefore, when the cache access this register from the P2 area of logical space or else make an appropr using the MMU so that this register is not cached.

* When address translation by the MMU does not apply, the address in paren should be used.

PTB7 to PTB0. Bits PB7DT to PB0DT correspond to pins PTB7 to PTB0. When the pi
is general output port, if the port is read the value of the corresponding PBDR bit is retu
directly. When the function is general input port, if the port is read the corresponding pi
read. Table 19.4 shows the function of PBDR.

PBDR is initialized to H'00 by a power-on reset. It retains its previous value in standby
sleep mode, and in a manual reset.

Table 19.4 Port B Data Register (PBDR) Read/Write Operations

PBnMD1	PBnMD0	Pin State	Read	Write
0	0	Other function (See table 18.1)	PBDR value	Value is written to PBDR, bu affect pin state
	1	Output	PBDR value	Write value is output from pi
1	0	Input (Pull-up MOS on)	Pin state	Value is written to PBDR, bu affect pin state
	1	Input (Pull-up MOS off)	Pin state	Value is written to PBDR, bu affect pin state

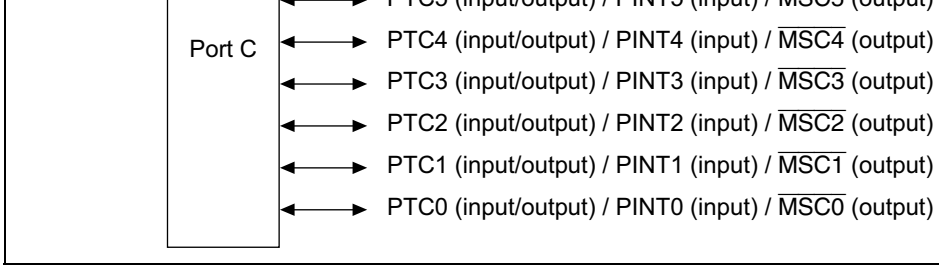


Figure 19.3 Port C

19.4.1 Register Description

Table 19.5 summarizes the port C register.

Table 19.5 Port C Register

Name	Abbreviation	R/W	Initial Value	Address	A
Port C data register	PCDR	R/W	H'00	H'04000124 (H'A4000124)*	8

Notes: This register is located in area 1 of physical space. Therefore, when the cache access this register from the P2 area of logical space or else make an appropr using the MMU so that this register is not cached.

* When address translation by the MMU does not apply, the address in paren should be used.

PTC7 to PTC0. Bits PC7DT to PC0DT correspond to pins PTC7 to PTC0. When the pi
is general output port, if the port is read, the value of the corresponding PCDR bit is ret
directly. When the function is general input port, if the port is read, the corresponding p
read. Table 19.6 shows the function of PCDR.

PCDR is initialized to H'00 by a power-on reset, after which the general input port func
up MOS on) is set as the initial pin function, and the corresponding pin levels are read.

PCDR retains its previous value in standby mode and sleep mode, and in a manual rese

Table 19.6 Port C Data Register (PCDR) Read/Write Operations

PCnMD1	PCnMD0	Pin State	Read	Write
0	0	Other function (see table 18.1)	PCDR value	Value is written to PCDR, bu affect pin state
	1	Output	PCDR value	Write value is output from pi
1	0	Input (Pull-up MOS on)	Pin state	Value is written to PCDR, bu affect pin state
	1	Input (Pull-up MOS off)	Pin state	Value is written to PCDR, bu affect pin state

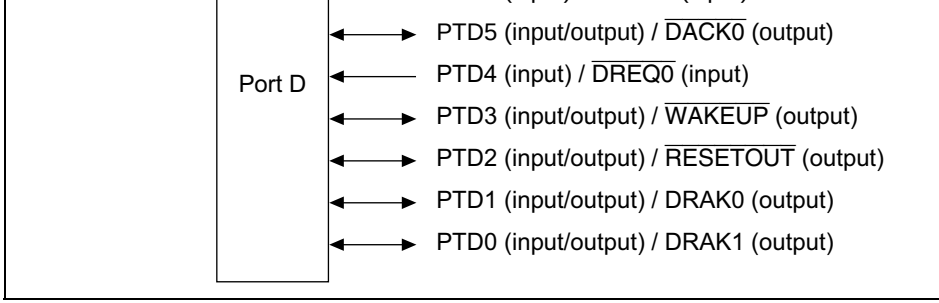


Figure 19.4 Port D

19.5.1 Register Description

Table 19.7 summarizes the port D register.

Table 19.7 Port D Register

Name	Abbreviation	R/W	Initial Value	Address	A
Port D data register	PDDR	R/W or R	B'0*0*0000	H'04000126 (H'A4000126)*1	8

Notes: This register is located in area 1 of physical space. Therefore, when the cache access this register from the P2 area of logical space or else make an appropr using the MMU so that this register is not cached.

* Means no value.

*1 When address translation by the MMU does not apply, the address in parer should be used.

The port D data register (PDDR) is a 6-bit readable/writable and 2-bit read-only registe
data for pins PTD7 to PTD0. Bits PD7DT to PD0DT correspond to pins PTD7 to PTD0
pin function is general output port, if the port is read, the value of the corresponding PD
returned directly. When the function is general input port, if the port is read, the corresp
level is read. Table 19.8 shows the function of PDDR.

PDDR is initialized to B'0*0*0000 by a power-on reset. After initialization, the general
function (pull-up MOS on) is set as the initial pin function, and the corresponding pin l
read from bits PD7DT—PD3DT, PD1DT, and PD0DT. PDDR retains its previous valu
standby mode and sleep mode, and in a manual reset.

Note that the low level is read if bits 6 and 4 are read except in general-purpose input.

Table 19.8 Port D Data Register (PDDR) Read/Write Operations

PDnMD1	PDnMD0	Pin State	Read	Write
0	0	Other function (see table 18.1)	PDDR value	Value is written to PDDR, bu affect pin state
	1	Output	PDDR value	Write value is output from pi
1	0	Input (Pull-up MOS on)	Pin state	Value is written to PDDR, bu affect pin state
	1	Input (Pull-up MOS off)	Pin state	Value is written to PDDR, bu affect pin state

(n = 0,

PDnMD1	PDnMD0	Pin State	Read	Write
0	0	Other function (see table 18.1)	Low level	Ignored (no effect on pin stat
	1	Reserved	Low level	Ignored (no effect on pin stat
1	0	Input (Pull-up MOS on)	Pin state	Ignored (no effect on pin stat
	1	Input (Pull-up MOS off)	Pin state	Ignored (no effect on pin stat

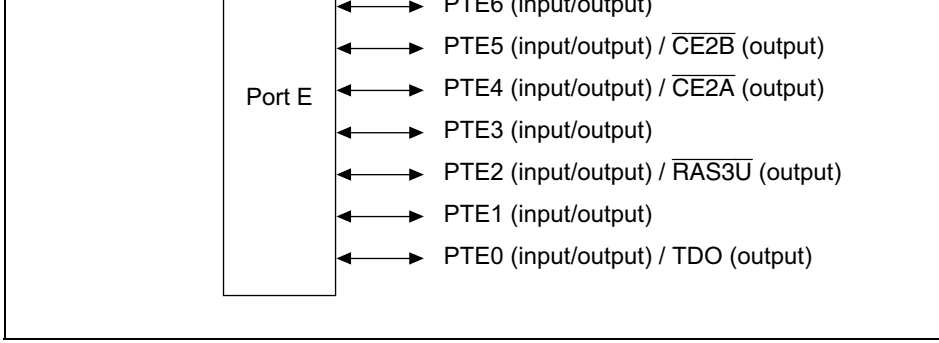


Figure 19.5 Port E

19.6.1 Register Description

Table 19.9 summarizes the port E register.

Table 19.9 Port E Register

Name	Abbreviation	R/W	Initial Value	Address	A
Port E data register	PEDR	R/W	H'00	H'04000128 (H'A4000128)*	8

Notes: This register is located in area 1 of physical space. Therefore, when the cache access this register from the P2 area of logical space or else make an appropr using the MMU so that this register is not cached.

* When address translation by the MMU does not apply, the address in parent should be used.

PTE7 to PTE0. Bits PE7DT to PE0DT correspond to pins PTE7 to PTE0. When the pin
general output port, if the port is read the value of the corresponding PEDR bit is return
When the function is general input port, if the port is read the corresponding pin level is
Table 19.10 shows the function of PEDR.

PEDR is initialized to H'00 by a power-on reset, after which the general input port funct
up MOS on) is set as the initial pin function, and the corresponding pin levels are read.
its previous value in standby mode and sleep mode, and in a manual reset.

Table 19.10 Port E Data Register (PEDR) Read/Write Operations

PEnMD1	PEnMD0	Pin State	Read	Write
0	0	Other function (see table 18.1)	PEDR value	Value is written to PEDR, bu affect pin state
	1	Output	PEDR value	Write value is output from pi
1	0	Input (Pull-up MOS on)	Pin state	Value is written to PEDR, bu affect pin state
	1	Input (Pull-up MOS off)	Pin state	Value is written to PEDR, bu affect pin state

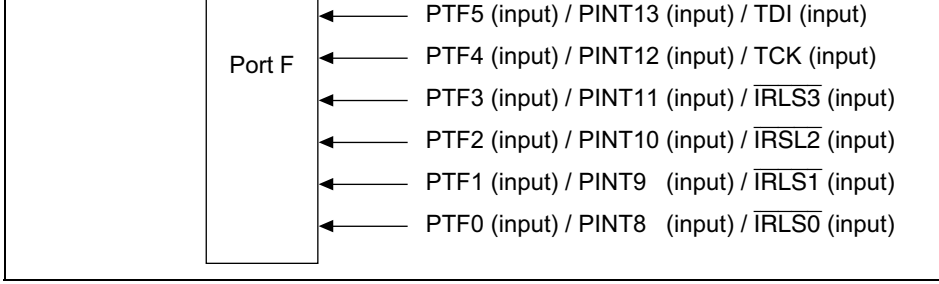


Figure 19.6 Port F

19.7.1 Register Description

Table 19.11 summarizes the port F register.

Table 19.11 Port F Register

Name	Abbreviation	R/W	Initial Value	Address	A
Port F data register	PFDR	R	H'**	H'0400012A (H'A400012A)*1	8

Notes: This register is located in area 1 of physical space. Therefore, when the cache access this register from the P2 area of logical space or else make an appropr using the MMU so that this register is not cached.

* Means no value.

*1 When address translation by the MMU does not apply, the address in parer should be used.

The port F data register (PFDR) is an 8-bit read-only register that stores data for pins P
PTF0. Bits PF7DT to PF0DT correspond to pins PTF7 to PTF0. When the function is g
input port, if the port is read the corresponding pin level is read. Table 19.12 shows the
PFDR.

PFDR is initialized by a power-on reset, after which the general input port function (pu
on) is set as the initial pin function, and the corresponding pin levels are read.

Table 19.12 Port F Data Register (PFDR) Read/Write Operations

PFnMD1	PFnMD0	Pin State	Read	Write
0	0	Other function (see table 18.1)	H'00	Ignored (no effect on pin stat
	1	Reserved	H'00	Ignored (no effect on pin stat
1	0	Input (Pull-up MOS on)	Pin state	Ignored (no effect on pin stat
	1	Input (Pull-up MOS off)	Pin state	Ignored (no effect on pin stat

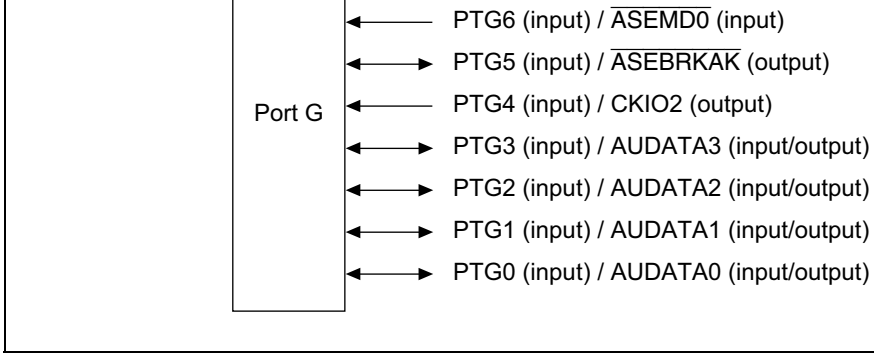


Figure 19.7 Port G

19.8.1 Register Description

Table 19.13 summarizes the port G register.

Table 19.13 Port G Register

Name	Abbreviation	R/W	Initial Value	Address	A
Port G data register	PGDR	R/W	H'**	H'0400012C (H'A400012C)*1	8

Notes: This register is located in area 1 of physical space. Therefore, when the cache access this register from the P2 area of logical space or else make an appropr using the MMU so that this register is not cached.

* Means no value.

*1 When address translation by the MMU does not apply, the address in parer should be used.

The port G data register (PGDR) is an 8-bit read-only register that stores data for pins P
PTG0. Bits PG7DT to PG0DT correspond to pins PTG7 to PTG0. When the function is
input port, if the port is read the corresponding pin level is read. Table 19.14 shows the
PGDR.

PGDR is initialized by a power-on reset, after which the general input port function (pu
on) is set as the initial pin function, and the corresponding pin levels are read.

Table 19.14 Port G Data Register (PGDR) Read/Write Operations

PGnMD1	PGnMD0	Pin State	Read	Write
0	0	Other function (see table 18.1)	H'00	Ignored (no effect on pin stat
	1	Reserved	H'00	Ignored (no effect on pin stat
1	0	Input (Pull-up MOS on)	Pin state	Ignored (no effect on pin stat
	1	Input (Pull-up MOS off)	Pin state	Ignored (no effect on pin stat

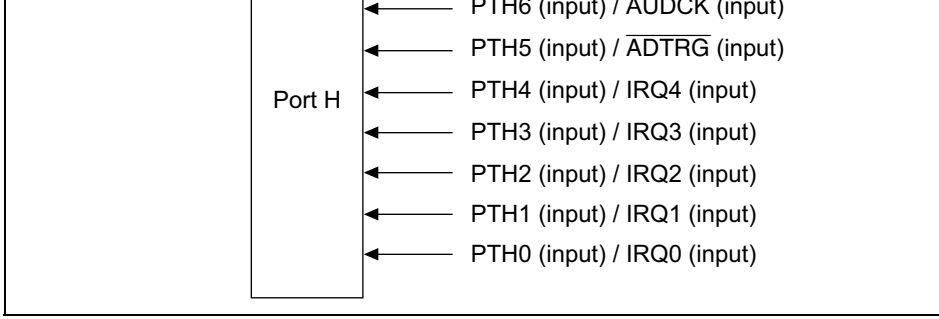


Figure 19.8 Port H

19.9.1 Register Description

Table 19.15 summarizes the port H register.

Table 19.15 Port H Register

Name	Abbreviation	R/W	Initial Value	Address
Port H data register	PHDR	R/W or R	B'0*******	H'0400012E (H'A400012E)*1

Notes: This register is located in area 1 of physical space. Therefore, when the cache
access this register from the P2 area of logical space or else make an appropr
using the MMU so that this register is not cached.

* Means no value.

*1 When address translation by the MMU does not apply, the address in parer
should be used.

The port H data register (PHDR) is a 1-bit readable/writable and 7-bit read-only registe
data for pins PTH7 to PTH0. Bits PH7DT to PH0DT correspond to pins PTH7 to PTH0
pin function is general output port, if the port is read, the value of the corresponding PH
returned directly. When the function is general input port, if the port is read, the corresp
level is read. Table 19.16 shows the function of PHDR.

PHDR is initialized to B'0******* by a power-on reset, after which the general input p
(pull-up MOS on) is set as the initial pin function, and the corresponding pin levels are
retains its previous value in standby mode and sleep mode, and in a manual reset.

Note that the low level is read if bits 6 to 0 are read except in general-purpose input.

Table 19.16 Port H Data Register (PHDR) Read/Write Operations

PHnMD1	PHnMD0	Pin State	Read	Write
0	0	Other function (see table 18.1)	PHDR value	Value is written to PHDR, bu affect pin state
	1	Output	PHDR value	Write value is output from pi
1	0	Input (Pull-up MOS on)	Pin state	Value is written to PHDR, bu affect pin state
	1	Input (Pull-up MOS off)	Pin state	Value is written to PHDR, bu affect pin state

PHnMD1	PHnMD0	Pin State	Read	Write
0	0	Other function (see table 18.1)	Low level	Ignored (no effect on pin sta
	1	Reserved	Low level	Ignored (no effect on pin sta
1	0	Input (Pull-up MOS on)	Pin state	Ignored (no effect on pin sta
	1	Input (Pull-up MOS off)	Pin state	Ignored (no effect on pin sta

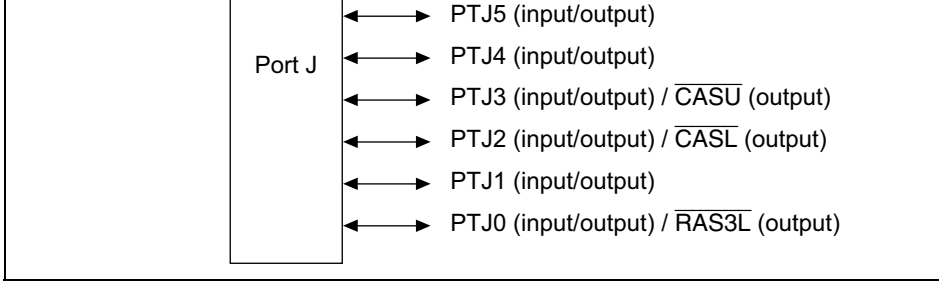


Figure 19.9 Port J

19.10.1 Register Description

Table 19.17 summarizes the port J register.

Table 19.17 Port J Register

Name	Abbreviation	R/W	Initial Value	Address	A
Port J data register	PJDR	R/W	H'00	H'04000130 (H'A4000130)*	8

Notes: This register is located in area 1 of physical space. Therefore, when the cache access this register from the P2 area of logical space or else make an appropr using the MMU so that this register is not cached.

* When address translation by the MMU does not apply, the address in paren should be used.

to PTJ0. Bits PJ7DT to PJ0DT correspond to pins PTJ7 to PTJ0. When the pin function
output port, if the port is read the value of the corresponding PJDR bit is returned direc
the function is general input port, if the port is read, the corresponding pin level is read.
19.18 shows the function of PJDR.

PJDR is initialized to H'00 by a power-on reset. It retains its previous value in software
mode and sleep mode, and in a manual reset.

Table 19.18 Port J Data Register (PJDR) Read/Write Operations

PJnMD1	PJnMD0	Pin State	Read	Write
0	0	Other function (see table 18.1)	PJDR value	Value is written to PJDR, but affect pin state
	1	Output	PJDR value	Write value is output from pi
1	0	Input (Pull-up MOS on)	Pin state	Value is written to PJDR, but affect pin state
	1	Input (Pull-up MOS off)	Pin state	Value is written to PJDR, but affect pin state

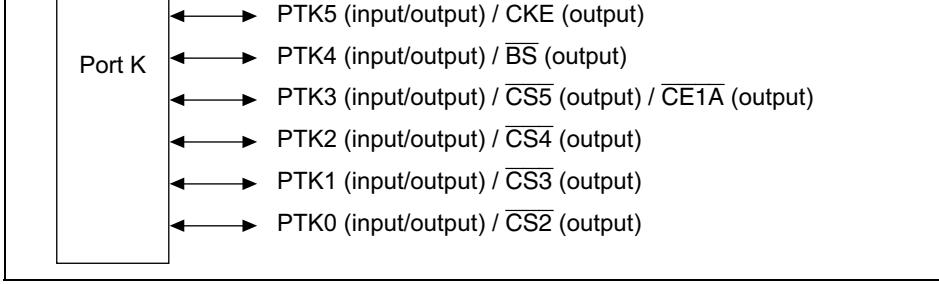


Figure 19.10 Port K

19.11.1 Register Description

Table 19.19 summarizes the port K register.

Table 19.19 Port K Register

Name	Abbreviation	R/W	Initial Value	Address	A
Port K data register	PKDR	R/W	H'00	H'04000132 (H'A4000132)*	8

Notes: This register is located in area 1 of physical space. Therefore, when the cache access this register from the P2 area of logical space or else make an appropr using the MMU so that this register is not cached.

* When address translation by the MMU does not apply, the address in parent should be used.

PTK7 to PTK0. Bits PK7DT to PK0DT correspond to pins PTK7 to PTK0. When the p
is general output port, if the port is read, the value of the corresponding PKDR bit is ret
directly. When the function is general input port, if the port is read, the corresponding p
read. Table 19.20 shows the function of PKDR.

PKDR is initialized to H'00 by a power-on reset. It retains its previous value in standby
sleep mode, and in a manual reset.

Table 19.20 Port K Data Register (PKDR) Read/Write Operations

PKnMD1	PKnMD0	Pin State	Read	Write
0	0	Other function (see table 18.1)	PKDR value	Value is written to PKDR, bu affect pin state
	1	Output	PKDR value	Write value is output from pi
1	0	Input (Pull-up MOS on)	Pin state	Value is written to PKDR, bu affect pin state
	1	Input (Pull-up MOS off)	Pin state	Value is written to PKDR, bu affect pin state

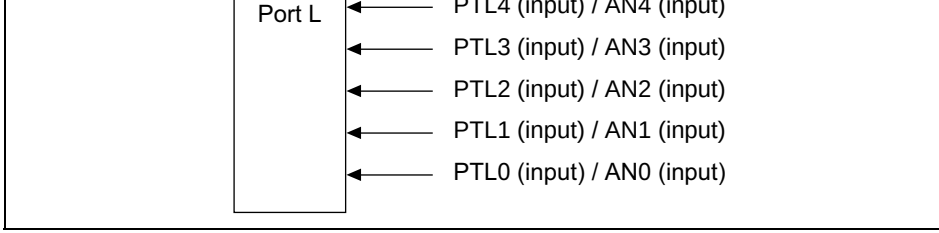


Figure 19.11 Port L

19.12.1 Register Description

Table 19.21 summarizes the port L register.

Table 19.21 Port L Register

Name	Abbreviation	R/W	Initial Value	Address	A
Port L data register	PLDR	R	H'00	H'04000134 (H'A4000134)*	8

Notes: This register is located in area 1 of physical space. Therefore, when the cache access this register from the P2 area of logical space or else make an appropr using the MMU so that this register is not cached.

* When address translation by the MMU does not apply, the address in paren should be used.

PTL0. Bits PL7DT to PL0DT correspond to pins PTL7 to PTL0. When the function is g
input port, if the port is read, the corresponding pin level is read. Table 19.22 shows the
of PLDR.

PKDR is initialized to H'00 by power-on reset. It retains its previous value in software s
mode and sleep mode, and in a manual reset.

The port L is also used as an analog pin, therefore does not have a pull-up MOS.

Table 19.22 Port L Data Register (PLDR) Read/Write Operation

PLnMD1	PLnMD0	Pin State	Read	Write
0	0	Other function (see table 18.1)	H'00	Ignored (no effect on pin stat
	1	Reserved	H'00	Ignored (no effect on pin stat
1	0	Input	Pin state	Ignored (no effect on pin stat
	1	Input	Pin state	Ignored (no effect on pin stat

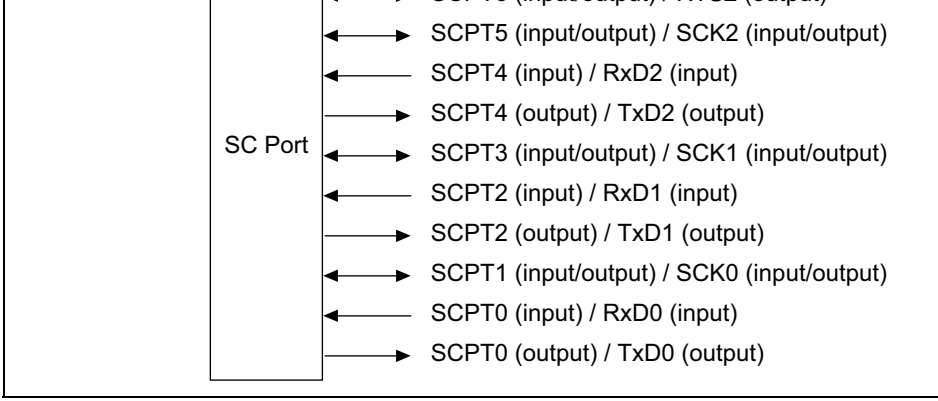


Figure 19.12 SC Port

19.13.1 Register Description

Table 19.23 summarizes the SC port register.

Table 19.23 SC Port Register

Name	Abbreviation	R/W	Initial Value	Address	A
SC Port data register	SCPDR	R/W or R	B'*0000000	H'04000136 (H'A4000136)*1	8

Notes: This register is located in area 1 of physical space. Therefore, when the cache access this register from the P2 area of logical space or else make an appropr using the MMU so that this register is not cached.

* Means no value.

*1 When address translation by the MMU does not apply, the address in parer should be used.

The SC port data register (SCPDR) is a 7-bit readable/writable and 1-bit read-only regis
stores data for pins SCPT7 to SCPT0. Bits SCP7DT to SCP0DT correspond to pins SC
SCPT0. When the pin function is general output port, if the port is read, the value of the
corresponding SCPDR bit is returned directly. When the function is general input port,
is read, the corresponding pin level is read. Table 19.24 shows the function of SCPDR.

SCPDR is initialized to B'*0000000 by a power-on reset. After initialization, the gener
port function (pull-up MOS on) is set as the initial pin function, and the corresponding
are read from bits SCP7DT—SCP5DT, SCP3DT, and SCP1DT. SCPDR retains its pre
value in standby mode and sleep mode, and in a manual reset.

Note that the low level is read if bit 7 is read except in general-purpose input.

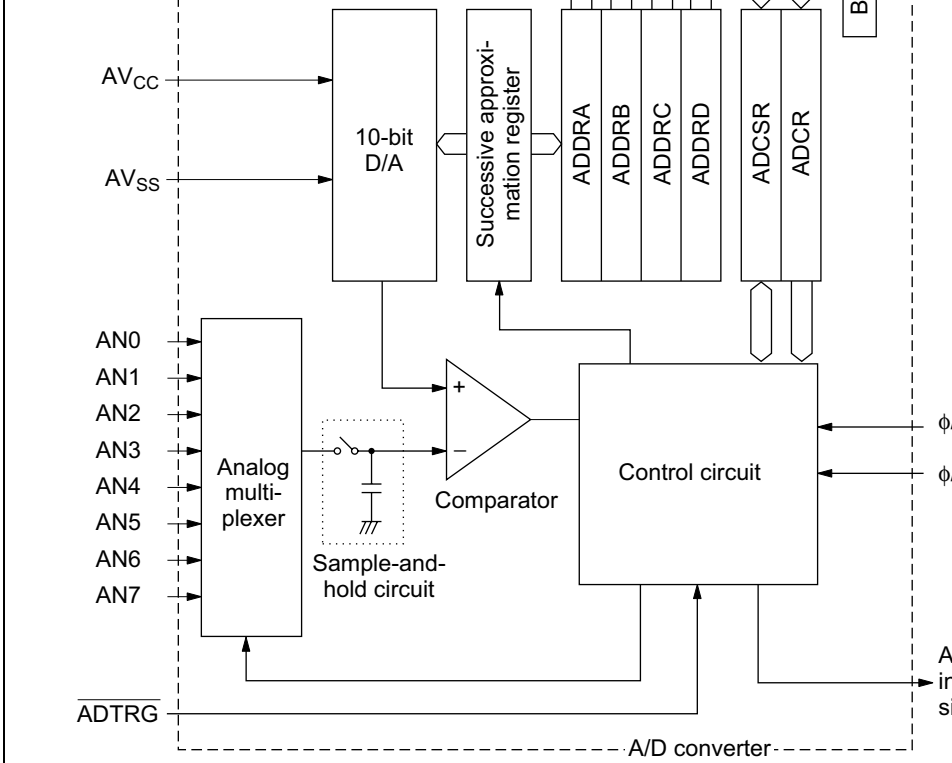
When the pin states of the RxD2 to RxD0 of the SCP4DT, SCP2DT, and SCP0DT bits
are read while the TE and RE bits in SCSCR are not cleared to 0, the RE bit in SCSCR
set to 1. When the RE bit is set to 1, the RxD pins become an input state and their pin st
read prior to the SCPCR setting.

		MOS off)		not affect pin state

SCPnMD1	SCPnMD0	Pin State	Read	Write
0	0	Other function (see table 18.1)	Low level	Ignored (no effect on pin
	1	Output	Low level	Ignored (no effect on pin
1	0	Input (Pull-up MOS on)	Pin state	Ignored (no effect on pin
	1	Input (Pull-up MOS off)	Pin state	Ignored (no effect on pin

A/D converter features are listed below.

- 10-bit resolution
- Eight input channels
- High-speed conversion
  - Conversion time: maximum 15 μs per channel (Pϕ = 33 MHz operation)
- Three conversion modes
  - Single mode: A/D conversion on one channel
  - Multi mode: A/D conversion on one to four channels
  - Scan mode: Continuous A/D conversion on one to four channels
- Four 16-bit data registers
  - A/D conversion results are transferred for storage into data registers correspond channels.
- Sample-and-hold function
- A/D conversion can be externally triggered
- A/D interrupt requested at the end of conversion
  - At the end of A/D conversion, an A/D end interrupt (ADI) can be requested.



Legend

ADCR: A/D control register
ADCSR: A/D control/status register
ADDRA: A/D data register A
ADDRB: A/D data register B
ADDRC: A/D data register C
ADDRD: A/D data register D

Figure 20.1 Block Diagram of A/D Converter

Analog power supply pin	AVcc	Input	Analog power supply
Analog ground pin	AVss	Input	Analog ground and referen
Analog input pin 0	AN0	Input	Group 0 analog inputs
Analog input pin 1	AN1	Input	
Analog input pin 2	AN2	Input	
Analog input pin 3	AN3	Input	
Analog input pin 4	AN4	Input	Group1 analog inputs
Analog input pin 5	AN5	Input	
Analog input pin 6	AN6	Input	
Analog input pin 7	AN7	Input	
A/D external trigger input pin	$\overline{\text{ADTRG}}$	Input	External trigger input for sta conversion

A/D data register AL	ADDRAL	R	H'00	H'04000082 (H'A4000082)*2	8
A/D data register BH	ADDRBH	R	H'00	H'04000084 (H'A4000084)*2	1
A/D data register BL	ADDRBL	R	H'00	H'04000086 (H'A4000086)*2	8
A/D data register CH	ADDRCH	R	H'00	H'04000088 (H'A4000088)*2	1
A/D data register CL	ADDRCL	R	H'00	H'0400008A (H'A400008A)*2	8
A/D data register DH	ADDRDH	R	H'00	H'0400008C (H'A400008C)*2	1
A/D data register DL	ADDRDL	R	H'00	H'0400008E (H'A400008E)*2	8
A/D control/status register	ADCSR	R/(W)*1	H'00	H'04000090 (H'A4000090)*2	8
A/D control register	ADCR	R/W	H'07	H'04000092 (H'A4000092)*2	8

Notes: These registers are located in area 1 of physical space. Therefore, when the ca
either access these registers from the P2 area of logical space or else make an
setting using the MMU so that these registers are not cached.

1. Only 0 can be written to bit 7, to clear the flag.
2. When address translation by the MMU does not apply, the address in parent
should be used.

Bit:							
Initial value:							
R/W:	R	R	R	R	R	R	R

Lower register: L

Bit:	7	6	5	4	3	2	1
	AD1	AD0	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R

n = A to D

The four A/D data registers (ADDRA to ADDRD) are 16-bit read-only registers that s
results of A/D conversion.

An A/D conversion produces 10-bit data, which is transferred for storage into the A/D
register corresponding to the selected channel. The upper 8 bits of the result are stored
byte (bits 7 to 0) of the A/D data register. The lower 2 bits are stored in the lower byte
6). Bits 5 to 0 of an A/D data register are reserved bits that are always read as 0. Table
indicates the pairings of analog input channels and A/D data registers.

The A/D data registers are initialized to H'0000 by a reset and in standby mode.

Table 20.3 Analog Input Channels and A/D Data Registers

Analog Input Channel		
Group 0	**Group 1**	**A/D Data Register**
AN0	AN4	ADDRA
AN1	AN5	ADDRB
AN2	AN6	ADDRC
AN3	AN7	ADDRD

ADCSR is an 8-bit readable/writable register that selects the mode and controls the A/D
ADCSR is initialized to H'00 by a reset and in standby mode.

Bit 7—A/D End Flag (ADF): Indicates the end of A/D conversion.

Bit 7: ADF	Description
0	[Clearing conditions] (In (1) Cleared by reading ADF while ADF = 1, then writing 0 to ADF (2) Cleared when DMAC is activated by ADI interrupt and ADDR is rea
1	[Setting conditions] (1) Single mode: A/D conversion ends (2) Multi mode: A/D conversion ends on all selected channels (3) Scan mode: A/D conversion ends on all selected channels

Bit 6—A/D Interrupt Enable (ADIE): Enables or disables the interrupt (ADI) request
end of A/D conversion. The ADIE bit should be set while the A/D conversion stops.

Bit 6: ADIE	Description
0	A/D end interrupt request (ADI) is disabled (In
1	A/D end interrupt request (ADI) is enabled

(3) Scan mode: A/D conversion starts and continues, cycling through channels, until ADST is cleared to 0 by software, by a reset, or by to standby mode

Bit 4—Multi Mode (MULTI): Selects single mode, multi mode or scan mode. For fu information on operation in these modes, see section 20.4, Operation.

Bit 4: MULTI	ADCR: Bit5: SCN	Description
0	0	Single mode
	1	
1	0	Multi mode
	1	Scan mode

Bit 3—Clock Select (CKS): Selects the A/D conversion time. Clear the ADST bit to changing the conversion time.

Bit 3:CKS	Description
0	Conversion time = 536 states (maximum)
1	Conversion time = 266 states (maximum)

		1	AN3	AN0 to AN3
1	0	0	AN4	AN4
		1	AN5	AN4, AN5
	1	0	AN6	AN4 to AN6
		1	AN7	AN4 to AN7

conversion. ADCR is initialized to H'07 by a reset and in standby mode.

Bit 7 and 6—Trigger Enable (TRGE1, TRGE0): Enables or disables external trigge
conversion.

The TRGE1 and TRGE0 bits should only be set when conversion is not in progress.

Bit 7: TRGE1	Bit 6: TRGE0	Description
0	0	A/D conversion does not start when an external tri
0	1	(I
1	0	A/D conversion starts at the falling edge of an inpu
1	1	the external trigger pin (ADTRG)

Bit 5—Scan Mode (SCN): Selects multi mode or scan mode when the MULTI bit is
the description of bit 4 in section 20.2.2, A/D Control/Status Register (ADCSR).

Bits 4 and 3—Reserved (RESVD1, RESVD2): These bits are always read as 0. The
should always be 0.

Bits 2 to 0—Reserved: These bits are always read as 1. The write value should alway

When reading an A/D data register, always read the upper byte before the lower byte. It to read only the upper byte, but if only the lower byte is read, the read value is not guar

Figure 20.2 shows the data flow for access to an A/D data register.

See section 20.7.3, Access Size and Read Data.

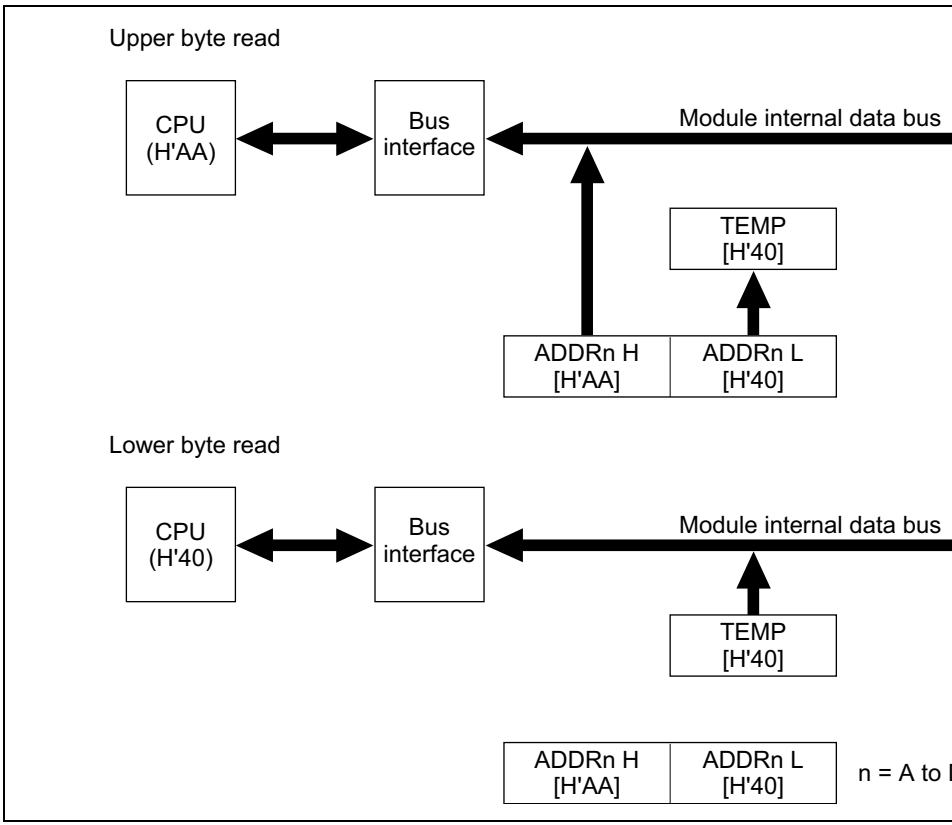


Figure 20.2 A/D Data Register Access Operation (Reading H'AA40)

ADST bit remains set to 1 during A/D conversion and is automatically cleared to 0 wh
conversion ends.

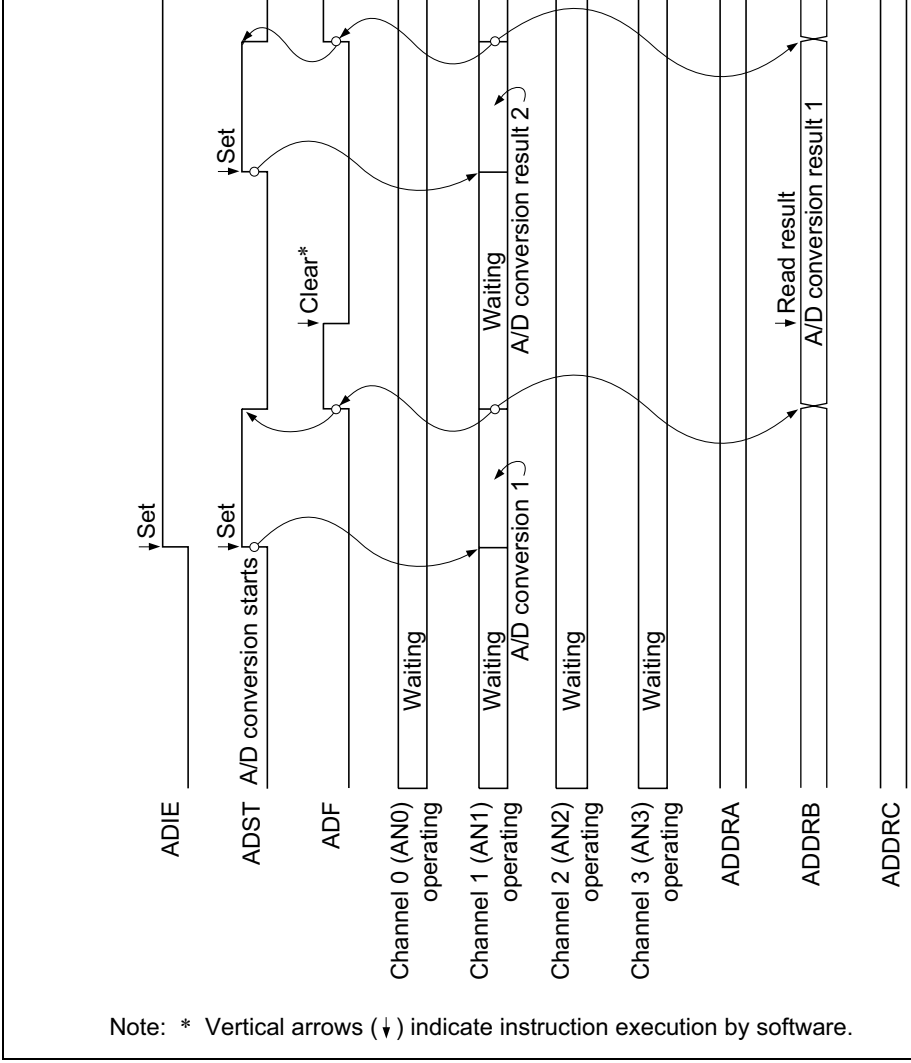
When conversion ends the ADF bit is set to 1. If the ADIE bit is also set to 1, an ADI
requested at this time. To clear the ADF bit to 0, first read ADF when ADF = 1, then
ADF bit.

When the mode or analog input channel must be switched during A/D conversion, to p
incorrect operation, first clear the ADST bit to 0 in ADCSR to halt A/D conversion. A
the necessary changes, set the ADST bit to 1 to start A/D conversion again. The ADS
set at the same time as the mode or channel is changed.

Typical operations when channel 1 (AN1) is selected in single mode are described nex

Figure 20.3 shows a timing diagram for this example. (The ADCSR register specifies
operation example.)

1. Single mode is selected (MULTI = 0), input channel AN1 is selected (CH2 = CH1
   1), the A/D interrupt is enabled (ADIE = 1), and A/D conversion is started (ADST
2. When A/D conversion is completed, the result is transferred into ADDRB. At the s
   the ADF flag is set to 1, the ADST bit is cleared to 0, and the A/D converter becon
3. Since ADF = 1 and ADIE = 1, an ADI interrupt is requested.
4. The A/D interrupt handling routine starts.
5. The routine reads ADCSR, then writes 0 to the ADF flag.
6. The routine reads and processes the conversion result (ADDRB = 0).
7. Execution of the A/D interrupt handling routine ends. Then, when the ADST bit is
   A/D conversion starts and steps 2 to 7 are executed.



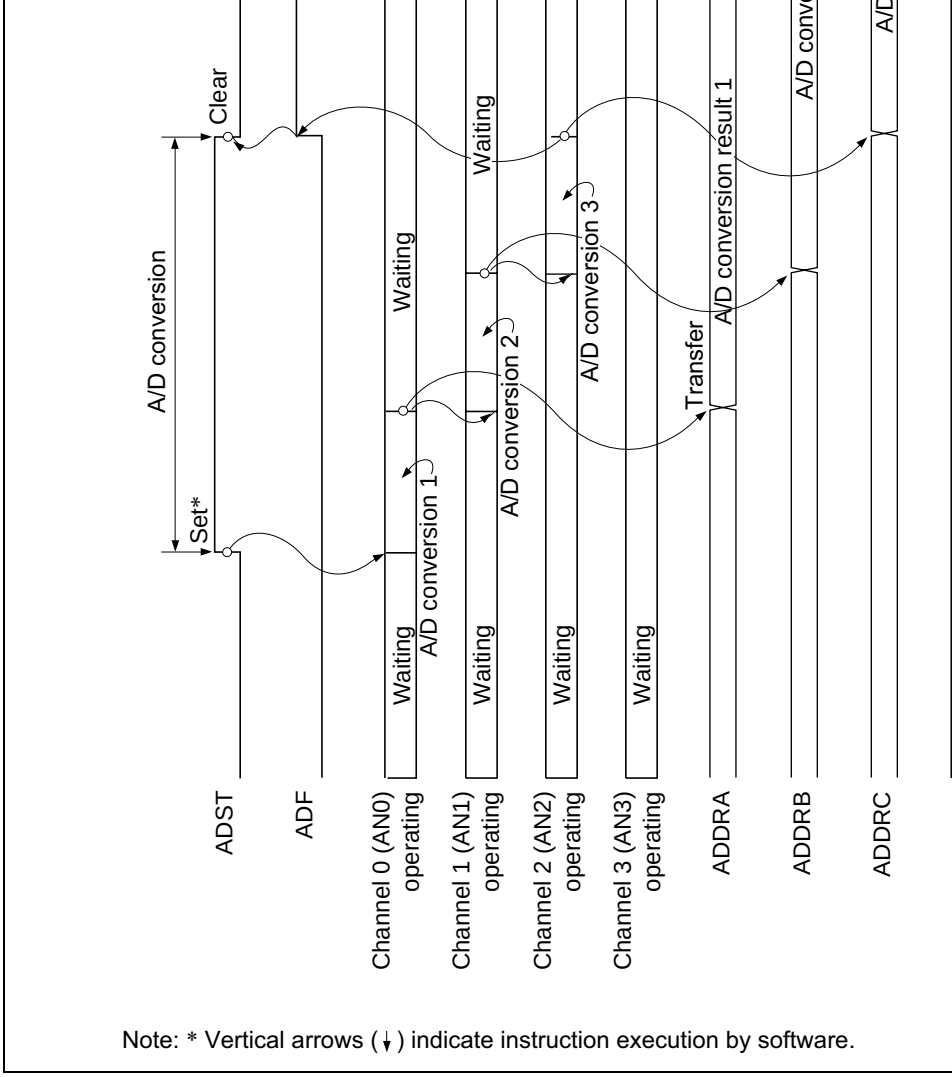
Note: * Vertical arrows (↓) indicate instruction execution by software.

Figure 20.3 Example of A/D Converter Operation (Single Mode, Channel 1 S

When the mode or analog input channel selection must be changed during A/D conver
prevent incorrect operation, first clear the ADST bit to 0 in ADCSR to halt A/D conve
making the necessary changes, set the ADST bit to 1. A/D conversion will start again
first channel in the group. The ADST bit can be set at the same time as the mode or ch
selection is changed.

Typical operations when three channels in group 0 (AN0 to AN2) are selected in scan
described next. Figure 20.4 shows a timing diagram for this example.

1. Multi mode is selected (MULTI = 1, SCN = 0), channel group 0 is selected (CH2 =
   input channels AN0 to AN2 are selected (CH1 = 1, CH0 = 0), and A/D conversion
   (ADST = 1).
2. When A/D conversion of the first channel (AN0) is completed, the result is transfe
   ADDRA. Next, conversion of the second channel (AN1) starts automatically.
3. Conversion proceeds in the same way through the third channel (AN2).
4. When conversion of all selected channels (AN0 to AN2) is completed, the ADF fla
   and ADST bit is cleared to 0. If the ADIE bit is set to 1, an ADI interrupt is reques
   time.



Note: * Vertical arrows (↓) indicate instruction execution by software.

Figure 20.4 Example of A/D Converter Operation (Multi Mode, Channels AN0 to AN2 Selected)

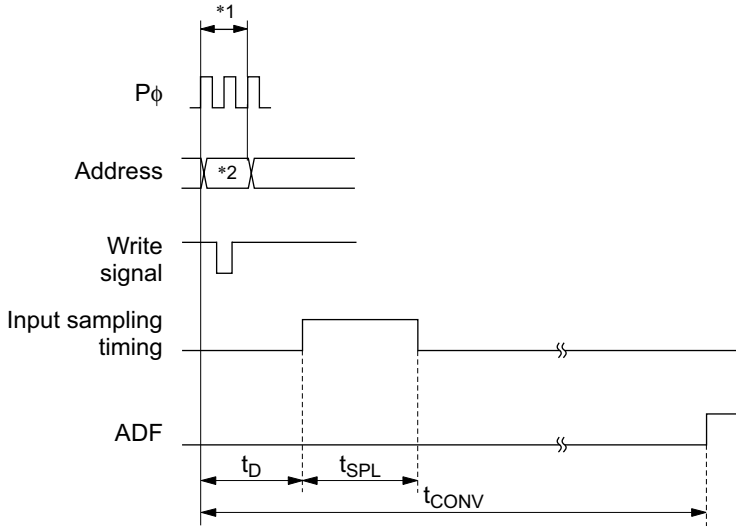
When the mode or analog input channel must be changed during analog conversion, to incorrect operation, first clear the ADST bit to 0 to halt A/D conversion. After making necessary changes, set the ADST bit to 1. A/D conversion will start again from the fi in the group. The ADST bit can be set at the same time as the mode or channel selecti changed.

Typical operations when three channels (AN0 to AN2) in group 0 are selected in scan described next. Figure 20.5 shows a timing diagram for this example.

1. Scan mode is selected (MULTI = 1, SCN = 1), channel group 0 is selected (CH2 = input channels AN0 to AN2 are selected (CH1 = 1, CH0 = 0), and A/D conversion (ADST = 1).
2. When A/D conversion of the first channel (AN0) is completed, the result is transfe ADDRA. Next, conversion of the second channel (AN1) starts automatically.
3. Conversion proceeds in the same way through the third channel (AN2).
4. When conversion of all the selected channels (AN0 to AN2) is completed, the AD to 1 and conversion of the first channel (AN0) starts again. If the ADIE bit is set t interrupt is requested at this time.
5. Steps 2 to 4 are repeated as long as the ADST bit remains set to 1. When the ADS cleared to 0, A/D conversion stops. After that, if the ADST bit is set to 1, A/D con starts again from the first channel (AN0).

Figure 20.5 Example of A/D Converter Operation (Scan Mode, Channels AN0 to AN2 Selected)

In multi mode and scan mode, the conversion time values given in table 20.4 apply to
conversion. In the second and subsequent conversions, the conversion time is fixed at
when CKS = 0 in ADCSR, or 256 states when CKS = 1. In both cases, the CKS bit sh
according to the Pϕ frequency so that the conversion time is within the range shown in
in section 23, Electrical Characteristics.



Legend

t_D	A/D conversion start delay
t_{SPL}	Input sampling time
t_{CONV}	A/D conversion time

Notes: 1. ADCSR write cycle
2. ADCSR address

Figure 20.6 A/D Conversion Timing

Note: * Values in the table are numbers of states (ϕ_{OSC}).

20.4.5 External Trigger Input Timing

A/D conversion can be externally triggered. When the TRGE1 and TRGE0 bits are set
ADCR, external trigger input is enabled at the $\overline{\text{ADTRG}}$ pin. A high-to-low transition at
$\overline{\text{ADTRG}}$ pin sets the ADST bit to 1 in ADCSR, starting A/D conversion. Other operatio
regardless of the conversion mode, are the same as if the ADST bit had been set to 1 by
Figure 20.7 shows the timing.

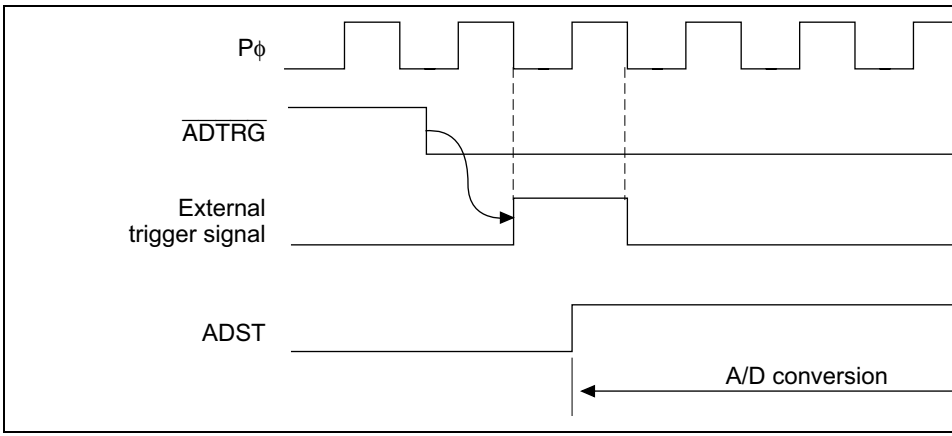


Figure 20.7 External Trigger Input Timing

is the deviation between the input analog value and the output digital value. It includes
following errors:

- Offset error
- Full-scale error
- Quantization error
- Nonlinearity error

These four error quantities are explained below with reference to figure 20.8. In the fi
bits of the A/D converter have been simplified to 3 bits.

Offset error is the deviation between actual and ideal A/D conversion characteristics w
digital output value changes from the minimum (zero voltage) 0000000000 (000 in th
000000001 (001 in the figure)(figure 20.8, item (1)). Full-scale error is the deviation b
actual and ideal A/D conversion characteristics when the digital output value changes
1111111110 (110 in the figure) to the maximum 1111111111 (111 in the figure)(figur
(2)). Quantization error is the intrinsic error of the A/D converter and is expressed as 1
(figure 20.8, item (3)). Nonlinearity error is the deviation between actual and ideal A/
characteristics between zero voltage and full-scale voltage (figure 20.8, item (4)). Not
not include offset, full-scale, or quantization error.

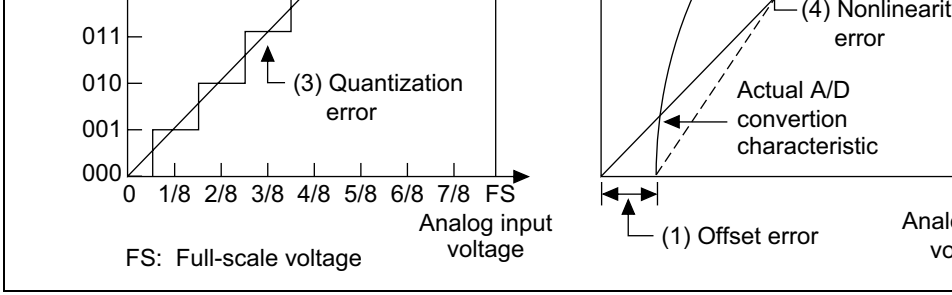


Figure 20.8 Definitions of A/D Conversion Accuracy

20.7 Usage Notes

When using the A/D converter, note the following points.

20.7.1 Setting Analog Input Voltage

- Analog Input Voltage Range: During A/D conversion, the voltages input to the anal pins ANn should be in the range $AV_{SS} \leq ANn \leq AV_{CC}$ (n = 0 to 7).
- Relationships of AV_{CC} and AV_{SS}: AV_{CC} and AV_{SS} should be related as follows: A $V_{CC} \pm 0.3$ V and $AV_{SS} = V_{SS}$.

20.7.2 Processing of Analog Input Pins

To prevent damage from voltage surges at the analog input pins (AN0 to AN7), connec protection circuit like the one shown in figure 20.9. The circuit shown also includes an suppress noise. This circuit is shown as an example; the circuit constants should be sele according to actual application conditions. Table 20.5 lists the analog input pin specific figure 20.10 shows an equivalent circuit diagram of the analog input ports.

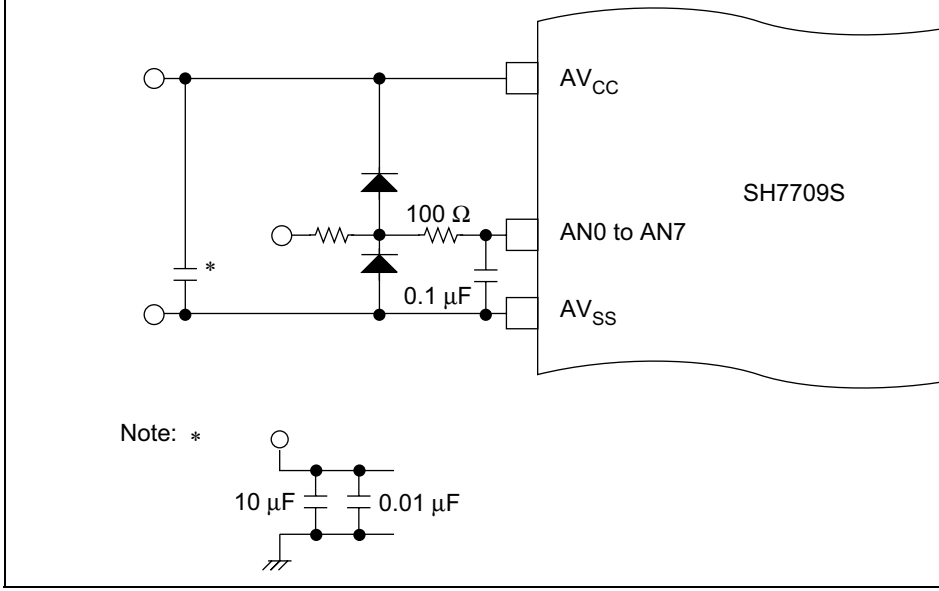


Figure 20.9 Example of Analog Input Protection Circuit

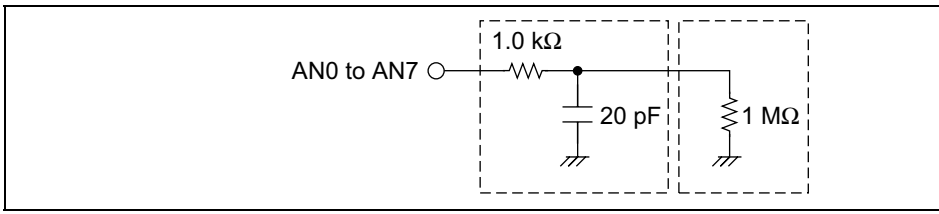


Figure 20.10 Analog Input Pin Equivalent Circuit

Access Size	Command	32 Bits (D31–D0)		16 Bits (D15–D0)		8 Bits
		Endian				
		Big	Little	Big	Little	Big
Byte access	MOV.L#ADDRAH,R9 MOV.B@R9,R8	FFFFFFFF	FFFFFFFF	FFFF	FFFF	FF
	MOV.L#ADDRAL,R9 MOV.B@R9,R8	C0C0C0C0	C0C0C0C0	C0C0	C0C0	C0
Word access	MOV.L#ADDRAH,R9 MOV.W@R9,R8	FFxxFFxx	FFxxFFxx	FFxx	FFxx	FF xx
	MOV.L#ADDRAL,R9 MOV.W@R9,R8	C0xxC0xx	C0xxC0xx	C0xx	C0xx	C0 xx
Longword access	MOV.L#ADDRAH,R9 MOV.L@R9,R8	FFxxC0xx	FFxxC0xx	Ffxx C0xx	C0xx FFxx	FF xx C0 xx

In this table: #ADDRAH .EQU H'04000080
#ADDRAL .EQU H'04000082
Values are shown in hexadecimal for the case where read data is output t
external device via R8.

D/A converter features are listed below.

- Eight-bit resolution
- Two output channels
- Conversion time: maximum 10 µs (with 20-pF capacitive load)
- Output voltage: 0 V to AVcc

21.1.2 Block Diagram

Figure 21.1 shows a block diagram of the D/A converter.

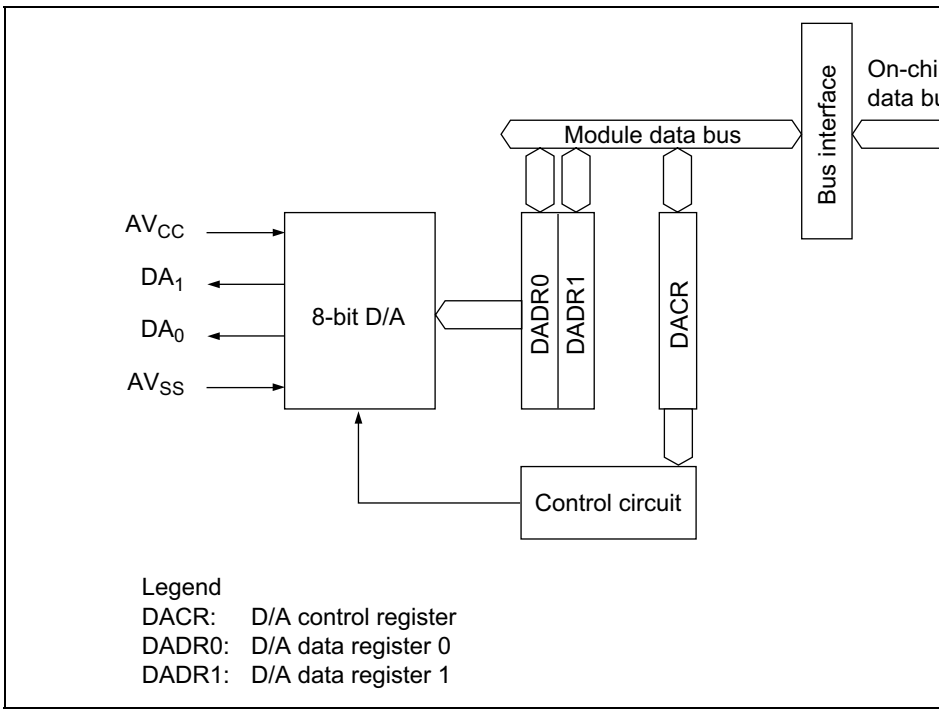


Figure 21.1 Block Diagram D/A Converter

Analog output pin 0	DA0	Output	Analog output, channel 0
Analog output pin 1	DA1	Output	Analog output, channel 1

21.1.4 Register Configuration

Table 21.2 summarizes the D/A converter's registers.

Table 21.2 D/A Converter Registers

Name	Abbreviation	R/W	Initial Value	Address
D/A data register 0	DADR0	R/W	H'00	H'040000 (H'A4000
D/A data register 1	DADR1	R/W	H'00	H'040000 (H'A4000
D/A control register	DACR	R/W	H'1F	H'040000 (H'A4000

Notes: These registers are located in area 1 of physical space. Therefore, when the ca
either access these registers from the P2 area of logical space or else make an
setting using the MMU so that these registers are not cached.

1. Lower 16 bits of the address
2. When address translation by the MMU does not apply, the address in parent
should be used.

The D/A data registers (DADR0 and DADR1) are 8-bit readable/writable registers tha
data to be converted. When analog output is enabled, the D/A data register values are
converted and output at the analog output pins.

The D/A data registers are initialized to H'00 by a reset.

21.2.2 D/A Control Register (DACR)

Bit:	7	6	5	4	3	2	1
	DAOE1	DAOE0	DAE	—	—	—	—
Initial value:	0	0	0	1	1	1	1
R/W:	R/W	R/W	R/W	R	R	R	R

DACR is an 8-bit readable/writable register that controls the operation of the D/A con
DACR is initialized to H'1F by a reset.

Bit 7—D/A Output Enable 1 (DAOE1): Controls D/A conversion and analog output

Bit 7: DAOE1	Description	
0	DA1 analog output is disabled	(In
1	Channel-1 D/A conversion and DA1 analog output are enabled	

Bit 6—D/A Output Enable 0 (DAOE0): Controls D/A conversion and analog output

Bit 6: DAOE0	Description	
0	DA0 analog output is disabled	(In
1	Channel-0 D/A conversion and DA0 analog output are enabled	

(In

0	1	0	D/A conversion is enabled in channel D/A conversion is disabled in channe
0	1	1	D/A conversion is enabled in channel
1	0	0	D/A conversion is disabled in channe D/A conversion is enabled in channel
1	0	1	D/A conversion is enabled in channel
1	1	—	D/A conversion is enabled in channel

When the DAE bit is set to 1, even if bits DAOE0 and DAOE1 in DACR and the ADST
ADCSR are cleared to 0, the same current is drawn from the analog power supply as du
and D/A conversion.

Bits 4 to 0—Reserved: Read-only bits, always read as 1.

An example of D/A conversion on channel 0 is given next. Timing is indicated in figu

1. Data to be converted is written in DADR0.
2. Bit DAOE0 is set to 1 in DACR. D/A conversion starts and DA0 becomes an outp converted result is output after the conversion time. The output value is (DADR0 c × AVcc. Output of this conversion result continues until the value in DADR0 is m the DAOE0 bit is cleared to 0.
3. If the DADR0 value is modified, conversion starts immediately, and the result is o the conversion time.
4. When the DAOE0 bit is cleared to 0, DA0 becomes an input pin.

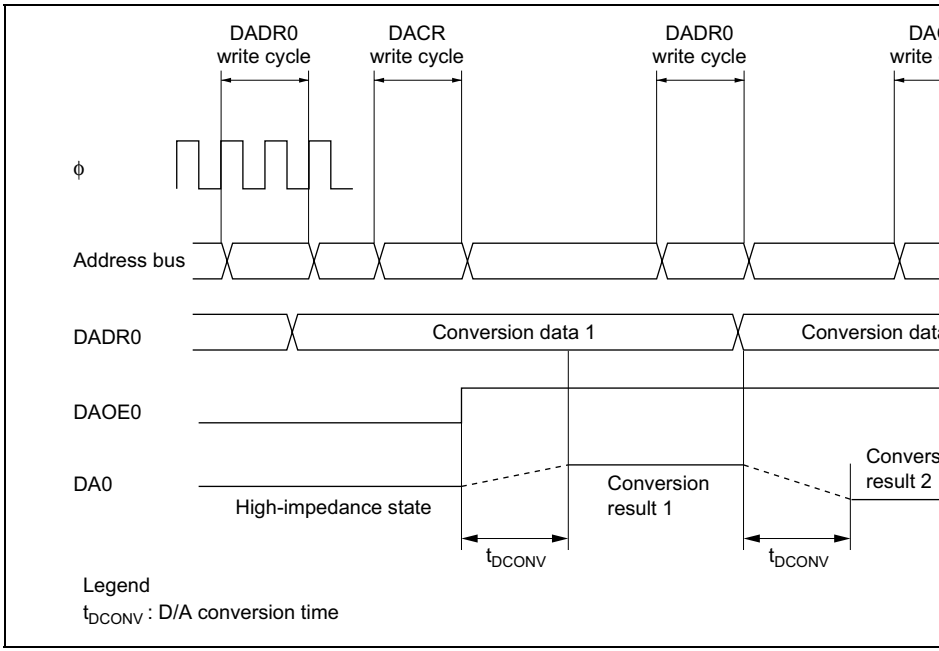


Figure 21.2 Example of D/A Converter Operation

22.2 User Debugging Interface (UDI)

The UDI (User debugging interface) performs on-chip debugging which is supported b
The UDI described here is a serial interface which is compatible with JTAG (Joint Tes
Group, IEEE Standard 1149.1 and IEEE Standard Test Access Port and Boundary-Sca
Architecture) specifications.

The UDI in the SH7709S supports a boundary scan mode, and is also used for emulato
connection.

When using an emulator, UDI functions should not be used. Refer to the emulator ma
method of connecting the emulator.

22.2.1 Pin Descriptions

TCK: UDI serial data input/output clock pin. Data is serially supplied to the UDI fro
input pin (TDI), and output from the data output pin (TDO), in synchronization with th

TMS: Mode select input pin. The state of the TAP control circuit is determined by ch
signal in synchronization with TCK. The protocol complies with the JTAG standard (
1149.1).

$\overline{\text{TRST}}$: UDI reset input pin. Input is accepted asynchronously with respect to TCK, an
the UDI is reset. See section 22.4.2, Reset Configuration, for more information.

TDI: UDI serial data input pin. Data transfer to the UDI is executed by changing this
synchronization with TCK.

TDO: UDI serial data output pin. Data output from the UDI is executed by reading th
synchronization with TCK.

$\overline{\text{ASEMD0}}$: ASE mode select pin. If a low level is input at the $\overline{\text{ASEMD0}}$ pin while the
pin is asserted, ASE mode is entered; if a high level is input, normal mode is entered.
on the user system alone, without an emulator and the UDI, hold this pin at high level.

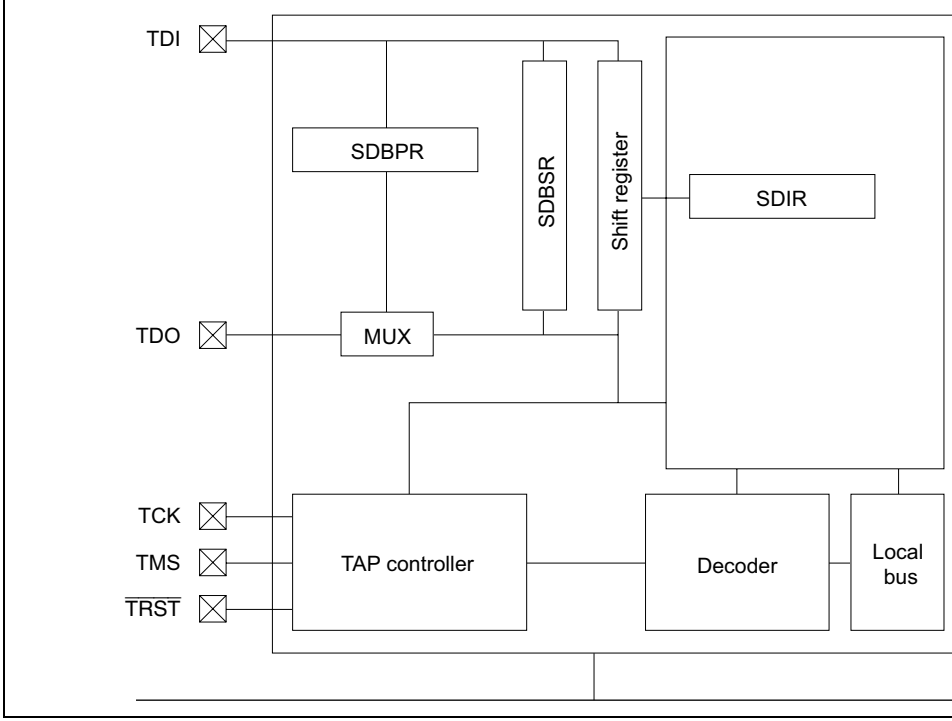


Figure 22.1 Block Diagram of UDI

22.3 Register Descriptions

The UDI has the following registers.

- SDBPR: Bypass register
- SDIR: Instruction register
- SDBSR: Boundary scan register

Note: * Initialized when $\overline{\text{TRST}}$ pin is low or when TAP is in the test-logic-reset state.

22.3.1 Bypass Register (SDBPR)

The bypass register is a 1-bit register that cannot be accessed by the CPU. When SDIR bypass mode, SDBPR is connected between UDI pins TDI and TDO.

22.3.2 Instruction Register (SDIR)

The instruction register (SDIR) is a 16-bit read-only register. The register is in bypass initial state. It is initialized by $\overline{\text{TRST}}$ assertion or in the TAP test-logic-reset state, and written to by the UDI irrespective of the CPU mode. Operation is not guaranteed if a r command is set in this register

Bit:	15	14	13	12	11	10	9
	TI3	TI2	TI1	TI0	—	—	—
Initial value:	1	1	1	1	1	1	1

Bit:	7	6	5	4	3	2	1
	—	—	—	—	—	—	—
Initial value:	1	1	1	1	1	1	1

Bits 15 to 12—Test Instruction Bits (TI3 to TI0): Cannot be written by the CPU.

0	1	1	1	UDI reset assert	
1	0	0	—	Reserved	
1	0	1	—	UDI interrupt	
1	1	0	—	Reserved	
1	1	1	0	Reserved	
1	1	1	1	Bypass mode	(In
0	0	0	1	Recovery from sleep	

Bits 11 to 0—Reserved: These bits are always read as 1.

22.3.3 Boundary Scan Register (SDBSR)

The boundary scan register (SDBSR) is a shift register, located on the PAD, for control input/output pins of the SH7709S.

Using the EXTEST and SAMPLE/PRELOAD commands, a boundary scan test conforn JTAG standard can be carried out. Table 22.3 shows the correspondence between the p LSI and boundary scan register bits.

334	D27/PTB3	IN	303	IRQ0/IRL0/PTH0
333	D26/PTB2	IN	302	IRQ1/$\overline{\text{IRL1}}$/PTH1
332	D25/PTB1	IN	301	IRQ2/$\overline{\text{IRL2}}$/PTH2
331	D24/PTB0	IN	300	IRQ3/$\overline{\text{IRL3}}$/PTH3
330	D23/PTA7	IN	299	IRQ4/PTH4
329	D22/PTA6	IN	298	D31/PTB7
328	D21/PTA5	IN	297	D30/PTB6
327	D20/PTA4	IN	296	D29/PTB5
326	D19/PTA3	IN	295	D28/PTB4
325	D18/PTA2	IN	294	D27/PTB3
324	D17/PTA1	IN	293	D26/PTB2
323	D16/PTA0	IN	292	D25/PTB1
322	D15	IN	291	D24/PTB0
321	D14	IN	290	D23/PTA7
320	D13	IN	289	D22/PTA6
319	D12	IN	288	D21/PTA5
318	D11	IN	287	D20/PTA4
317	D10	IN	286	D19/PTA3
316	D9	IN	285	D18/PTA2
315	D8	IN	284	D17/PTA1
314	D7	IN	283	D16/PTA0
313	D6	IN	282	D15
312	D5	IN	281	D14
311	D4	IN	280	D13
310	D3	IN	279	D12
309	D2	IN	278	D11

270	D3	OUT	240	D5
269	D2	OUT	239	D4
268	D1	OUT	238	D3
267	D0	OUT	237	D2
266	D31/PTB7	Control	236	D1
265	D30/PTB6	Control	235	D0
264	D29/PTB5	Control	234	$\overline{BS}$/PTK4
263	D28/PTB4	Control	233	$\overline{WE2}$/DQMUL/$\overline{ICIORD}$/ PTK6
262	D27/PTB3	Control	232	$\overline{WE3}$/DQMUU/$\overline{ICIORD}$/ PTK7
261	D26/PTB2	Control	231	$\overline{AUDSYNC}$/PTE7
260	D25/PTB1	Control	230	$\overline{CS2}$/PTK0
259	D24/PTB0	Control	229	$\overline{CS3}$/PTK1
258	D23/PTA7	Control	228	$\overline{CS4}$/PTK2
257	D22/PTA6	Control	227	$\overline{CS5}$/$\overline{CE1A}$/PTK3
256	D21/PTA5	Control	226	$\overline{CE2A}$/PTE4
255	D20/PTA4	Control	225	$\overline{CE2B}$/PTE5
254	D19/PTA3	Control	224	A0
253	D18/PTA2	Control	223	A1
252	D17/PTA1	Control	222	A2
251	D16/PTA0	Control	221	A3
250	D15	Control	220	A4
249	D14	Control	219	A5
248	D13	Control	218	A6

210	A14	OUT	180	A2
209	A15	OUT	179	A3
208	A16	OUT	178	A4
207	A17	OUT	177	A5
206	A18	OUT	176	A6
205	A19	OUT	175	A7
204	A20	OUT	174	A8
203	A21	OUT	173	A9
202	A22	OUT	172	A10
201	A23	OUT	171	A11
200	A24	OUT	170	A12
199	A25	OUT	169	A13
198	$\overline{\text{BS}}$/PTK4	OUT	168	A14
197	$\overline{\text{RD}}$	OUT	167	A15
196	$\overline{\text{WE0}}$/DQMLL	OUT	166	A16
195	$\overline{\text{WE1}}$/DQMLU/$\overline{\text{WE}}$	OUT	165	A17
194	$\overline{\text{WE2}}$/DQMUL/$\overline{\text{ICIORD}}$/ PTK6	OUT	164	A18
193	$\overline{\text{WE3}}$/DQMUU/$\overline{\text{ICIOWR}}$/ PTK7	OUT	163	A19
192	RD/$\overline{\text{WR}}$	OUT	162	A20
191	$\overline{\text{AUDSYNC}}$/PTE7	OUT	161	A21
190	$\overline{\text{CS0}}$/MCS0	OUT	160	A22
189	$\overline{\text{CS2}}$/PTK0	OUT	159	A23
188	$\overline{\text{CS3}}$/PTK1	OUT	158	A24

151	$\overline{WE3}$/DQMUU/$\overline{ICIOWR}$/PTK7	Control	121	AUDATA3/PTG3
150	RD/$\overline{WR}$	Control	120	AUDATA2/PTG2
149	$\overline{AUDSYNC}$/PTE7	Control	119	AUDATA1/PTG1
148	$\overline{CS0}$/MCS0	Control	118	AUDATA0/PTG0
147	$\overline{CS2}$/PTK0	Control	117	$\overline{ADTRG}$/PTH5
146	$\overline{CS3}$/PTK1	Control	116	$\overline{IRLS3}$/PTF3/PINT11
145	$\overline{CS4}$/PTK2	Control	115	$\overline{IRLS2}$/PTF2/PINT10
144	$\overline{CS5}$/$\overline{CE1A}$/PTK3	Control	114	$\overline{IRLS1}$/PTF1/PINT9
143	$\overline{CS6}$/$\overline{CE1B}$	Control	113	$\overline{IRLS0}$/PTF0/PINT8
142	$\overline{CE2A}$/PTE4	Control	112	MD0
141	$\overline{CE2B}$/PTE5	Control	111	CKE/PTK5
140	CKE/PTK5	IN	110	$\overline{RAS3L}$/PTJ0
139	$\overline{RAS3L}$/PTJ0	IN	109	PTJ1
138	PTJ1	IN	108	$\overline{CASL}$/PTJ2
137	$\overline{CASL}$/PTJ2	IN	107	$\overline{CASU}$/PTJ3
136	$\overline{CASU}$/PTJ3	IN	106	PTJ4
135	PTJ4	IN	105	PTJ5
134	PTJ5	IN	104	DACK0/PTD5
133	DACK0/PTD5	IN	103	DACK1/PTD7
132	DACK1/PTD7	IN	102	PTE6
131	PTE6	IN	101	PTE3
130	PTE3	IN	100	$\overline{RAS3U}$/PTE2
129	$\overline{RAS3U}$/PTE2	IN	99	PTE1
128	PTE1	IN	98	$\overline{BACK}$

90	PTJ1	Control	58	RxD1/SCPT2	
89	$\overline{\text{CASL}}$/PTJ2	Control	57	$\overline{\text{CTS2}}$/IRQ5/SCPT7	
88	$\overline{\text{CASU}}$/PTJ3	Control	56	$\overline{\text{MCS7}}$/PTC7/PINT7	
87	PTJ4	Control	55	$\overline{\text{MCS6}}$/PTC6/PINT6	
86	PTJ5	Control	54	$\overline{\text{MCS5}}$/PTC5/PINT5	
85	DACK0/PTD5	Control	53	$\overline{\text{MCS4}}$/PTC4/PINT4	
84	DACK1/PTD7	Control	52	$\overline{\text{MCS3}}$/PTC3/PINT3	
83	PTE6	Control	51	$\overline{\text{MCS2}}$/PTC2/PINT2	
82	PTE3	Control	50	$\overline{\text{MCS1}}$/PTC1/PINT1	
81	$\overline{\text{RAS3U}}$/PTE2	Control	49	$\overline{\text{MCS0}}$/PTC0/PINT0	
80	PTE1	Control	48	MD3	
79	$\overline{\text{BACK}}$	Control	47	MD4	
78	$\overline{\text{ASEBRKAK}}$/PTG5	Control	46	MD5	
77	AUDATA3/PTG3	Control	45	STATUS0/PTJ6	
76	AUDATA2/PTG2	Control	44	STATUS1/PTJ7	
75	AUDATA1/PTG1	Control	43	TCLK/PTH7	
74	AUDATA0/PTG0	Control	42	$\overline{\text{IRQOUT}}$	
73	STATUS0/PTJ6	IN	41	TxD0/SCPT0	
72	STATUS1/PTJ7	IN	40	SCK0/SCPT1	
71	TCLK/PTH7	IN	39	TxD1/SCPT2	
70	SCK0/SCPT1	IN	38	SCK1/SCPT3	
69	SCK1/SCPT3	IN	37	TxD2/SCPT4	
68	SCK2/SCPT5	IN	36	SCK2/SCPT5	
67	$\overline{\text{RTS2}}$/SCPT6	IN	35	RTS2/SCPT6	
66	RxD0/SCPT0	IN	34	$\overline{\text{MCS7}}$/PTC7/PINT7	

26	$\overline{\text{MCS1}}$/PTC1/PINT1	OUT	8	$\overline{\text{MCS4}}$/PTC4/PINT4
25	$\overline{\text{MCS0}}$/PTC0/PINT0	OUT	7	$\overline{\text{WAKEUP}}$/PTD3
24	DRAK0/PTD1	OUT	6	$\overline{\text{RESETOUT}}$/PTD2
23	DRAK1/PTD0	OUT	5	$\overline{\text{MCS3}}$/PTC3/PINT3
22	STATUS0/PTJ6	Control	4	$\overline{\text{MCS2}}$/PTC2/PINT2
21	STATUS1/PTJ7	Control	3	$\overline{\text{MCS1}}$/PTC1/PINT1
20	TCLK/PTH7	Control	2	$\overline{\text{MCS0}}$/PTC0/PINT0
19	$\overline{\text{IRQOUT}}$	Control	1	DRAK0/PTD1
18	TxD0/SCPT0	Control	0	DRAK1/PTD0
17	SCK0/SCPT1	Control	to TDO	
16	TxD1/SCPT2	Control		

Note: Control is an active-low signal.
When Control is driven low, the corresponding pin is driven by the value of OUT.

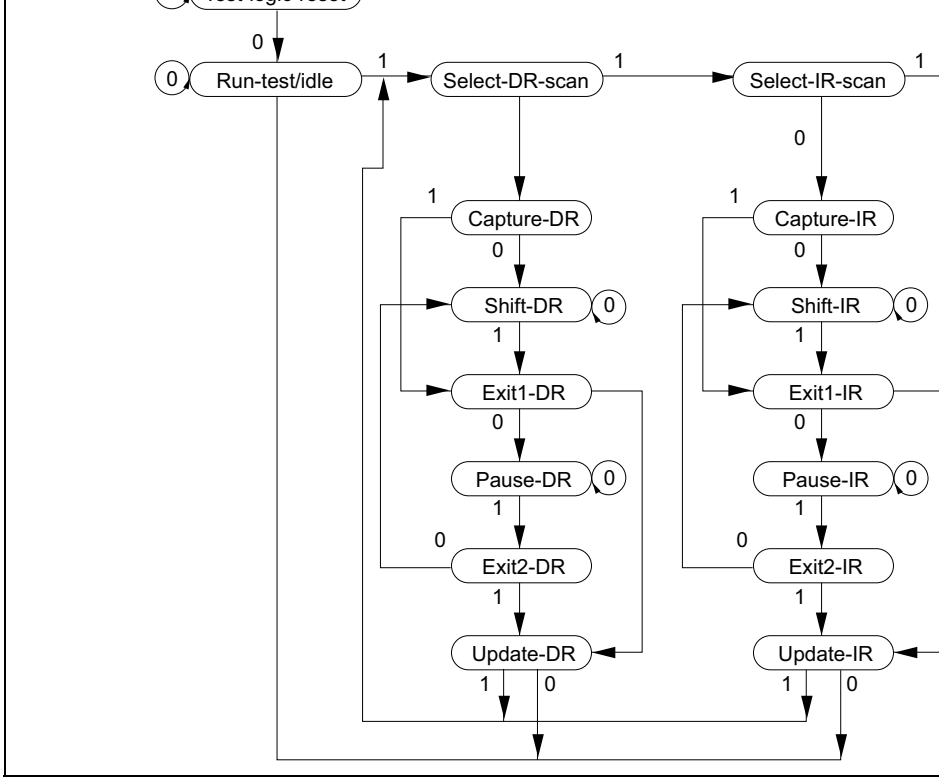


Figure 22.2 TAP Controller State Transitions

Note: The transition condition is the TMS value at the rising edge of TCK. The TDI sampled at the rising edge of TCK; shifting occurs at the falling edge of TCK. value changes at the TCK falling edge. The TDO is at high impedance, except DR (shift-SR) and shift-IR states. During the change to $\overline{\text{TRST}}$ = 0, there is a tr test-logic-reset asynchronously with TCK.

Low-level	Low-level	Low-level	Reset hold[*2]
		High-level	ASE user mode[*3]: Normal reset ASE break mode[*3]: $\overline{\text{RESETP}}$ asse masked
	High-level	Low-level	UDI reset only
		High-level	Normal operation

Notes: 1. Selects main chip mode or ASE mode
$\overline{\text{ASEMD0}}$ = H, normal mode
$\overline{\text{ASEMD0}}$ = L, ASE mode
Set $\overline{\text{ASEMD0}}$ = H when using on the user system alone, without an emulator
UDI.

2. In ASE mode, reset hold is enabled by driving the $\overline{\text{RESETP}}$ and $\overline{\text{TRST}}$ pins l
constant cycle. In this state, the CPU does not start up, even if $\overline{\text{RESETP}}$ is d
When $\overline{\text{TRST}}$ is driven high, UDI operation is enabled, but the CPU does not
The reset hold state is cancelled by the following:
   - Boot request from UDI
   - Another $\overline{\text{RESETP}}$ assert (power-on reset)

3. There are two ASE modes, one for executing software in the emulator's firm
break mode) and one for executing user software (ASE user mode).

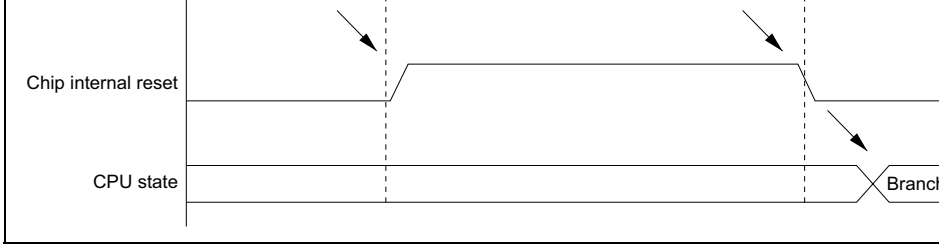


Figure 22.3 UDI Reset

22.4.4 UDI Interrupt

The UDI interrupt function generates an interrupt by setting a command from the UDI SDIR. An UDI interrupt is a general exception/interrupt operation, resulting in a branc address based on the VBR value plus offset, and with return by the RTE instruction. T request has a fixed priority level of 15.

UDI interrupts are not accepted in sleep mode or standby mode.

22.4.5 Bypass

The JTAG-based bypass mode for the UDI pins can be selected by setting a command UDI in SDIR.

22.4.6 Using UDI to Recover from Sleep Mode

It is possible to recover from sleep mode by setting a command (0001) from the UDI i

BYPASS: The BYPASS instruction is an essential standard instruction that operates th
register. This instruction shortens the shift path to speed up serial data transfer involvi
chips on the printed circuit board. While this instruction is executing, the test circuit ha
on the system circuits. The instruction code is 1111.

SAMPLE/PRELOAD: The SAMPLE/PRELOAD instruction inputs values from this
internal circuitry to the boundary scan register, outputs values from the scan path, and l
onto the scan path. When this instruction is executing, this LSI's input pin signals are t
directly to the internal circuitry, and internal circuit values are directly output externall
output pins. This LSI's system circuits are not affected by execution of this instruction.
instruction code is 0100.

In a SAMPLE operation, a snapshot of a value to be transferred from an input pin to the
circuitry, or a value to be transferred from the internal circuitry to an output pin, is latc
boundary scan register and read from the scan path. Snapshot latching is performed in
synchronization with the rise of TCK in the Capture-DR state. Snapshot latching does
normal operation of this LSI.

In a PRELOAD operation, an initial value is set in the parallel output latch of the bound
register from the scan path prior to the EXTEST instruction. Without a PRELOAD ope
when the EXTEST instruction was executed an undefined value would be output from
pin until completion of the initial scan sequence (transfer to the output latch) (with the
instruction, the parallel output latch value is constantly output to the output pin).

EXTEST: This instruction is provided to test external circuitry when this LSI is mount
printed circuit board. When this instruction is executed, output pins are used to output
(previously set by the SAMPLE/PRELOAD instruction) from the boundary scan regist
printed circuit board, and input pins are used to latch test results into the boundary scan
from the printed circuit board. If testing is carried out by using the EXTEST instructio
the Nth test data is scanned-in when test data (N-1) is scanned out.

2. Boundary scan mode does not cover reset-related signals ($\overline{\text{RESETP}}$, $\overline{\text{RESETM}}$, CA
3. Boundary scan mode does not cover UDI-related signals (TCK, TDI, TDO, TMS,
4. When a boundary scan test is carried out, ensure that the CKIO clock operates con
   The CKIO frequency range is as follows:
   Minimum: 1 MHz
   Maximum: Maximum frequency for respective clock mode specified in the CPG s
   Set pins MD[2:0] to the clock mode to be used.
   After powering on, wait for the CKIO clock to stabilize before performing a bound
   test.
5. Fix the $\overline{\text{RESETP}}$ pin low.
6. Fix the CA pin high, and the $\overline{\text{ASEMD0}}$ pin low.

22.6 Usage Notes

1. An UDI command other than an UDI interrupt, once set, will not be modified as lo
   another command is not re-issued from the UDI. An UDI interrupt command, hov
   be changed to a bypass command once set.
2. Because chip operations are suspended in standby mode, UDI commands are not a
   However, the TAP controller remains in operation at this time.
3. The UDI is used for emulator connection. Therefore, UDI functions cannot be use
   using an emulator.

22.7 Advanced User Debugger (AUD)

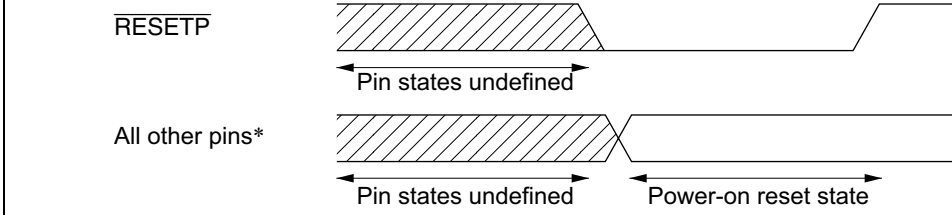
The AUD is a function exclusively for use by an emulator. Refer to the User's Manua
relevant emulator for details of the AUD.

Power supply voltage (I/O)	VccQ	–0.3 to 4.2	V
Power supply voltage (internal)	Vcc Vcc – PLL1 Vcc – PLL2 Vcc – RTC	–0.3 to 2.5	V
Input voltage (except port L)	Vin	–0.3 to VccQ + 0.3	V
Input voltage (port L)	Vin	–0.3 to AVcc + 0.3	V
Analog power-supply voltage	AVcc	–0.3 to 4.6	V
Analog input voltage	V_{AN}	–0.3 to AVcc + 0.3	V
Operating temperature	Topr	–20 to 75	°C
Storage temperature	Tstr	–55 to 125	°C

Caution: Operating the chip in excess of the absolute maximum rating may result in p damage.

- Order of turning on 1.7 V/1.8 V/1.9 V/2.0 V power (Vcc, Vcc-PLL1, Vcc-PLL2, V and 3.3 V power (VccQ, AVcc):
  1. First turn on the 3.3 V power, then turn on the 1.7 V/1.8 V/1.9 V/2.0 V power v This interval should be as short as possible.
  2. Until voltage is applied to all power supplies, a low level is input at the $\overline{\text{RESET}}$ CKIO has operated for a maximum of 4 clock cycles, internal circuits remain u so pin states are also undefined. The system design must ensure that these unde do not cause erroneous system operation. Note that the $\overline{\text{RESETP}}$ pin cannot rec level signal while a low level signal is being input to the CA pin.

  Waveforms at power-on are shown in the following figure.



Note: * Except power/GND, clock related, and analog pins

Power-On Sequence

- Power-off order
  1. In the reverse order of powering-on, first turn off the 1.7 V/1.8 V/1.9 V/2.0 V po
     turn off the 3.3 V power within 1 ms. This interval should be as short as possible
  2. Pin states are undefined while only the 1.7 V/1.8 V/1.9 V/2.0 V power is off. Th
     design must ensure that these undefined states do not cause erroneous system op

Power supply voltage		VccQ	3.0	3.3	3.6	V	
		Vcc,	1.85	2.00	2.15		200 MHz mode
		Vcc-PLL1,	1.75	1.90	2.05		167 MHz mode
		Vcc-PLL2,	1.65	1.80	2.05		133 MHz mode
		Vcc-RTC	1.55	1.70	1.95		100 MHz mode
Current dissipation	Normal operation	Icc	—	410	680	mA	Vcc = 2.0 V, Iϕ
				330	540		Vcc = 1.9 V, Iϕ
			—	250	410		Vcc = 1.8 V, Iϕ
			—	190	310		Vcc = 1.7 V, Iϕ
		IccQ	—	20	40		VccQ = 3.3 V, I
	Sleep mode*1	Icc	—	15	30		*1 When there i external bus cy than the refresh
		IccQ	—	10	20		Vcc = 1.9 V, VccQ = 3.3 V Bϕ = 33MHz
	Standby mode	Icc	—	40	120	μA	Ta = 25°C (RT VccQ = 3.3 V, Vcc = 1.55 V to
		IccQ	—	10	30		
		Icc	—	290	900		Ta = 25°C (RT Crystal is not u VccQ = 3.3 V, Vcc = 1.55 V to
		IccQ	—	10	30		

	PINT0, ASEMD0, ADTRG, TRST, EXTAL, CKIO, RxD1, CA				
	EXTAL2		—	—	—	When not conne crystal oscillator Vcc.
	Port L		2.0	—	AVcc + 0.3	
	Other input pins		2.0	—	VccQ + 0.3	

	PINT0, $\overline{\text{ASEMD0}}$, $\overline{\text{ADTRG}}$, $\overline{\text{TRST}}$, EXTAL, CKIO, RxD1, CA						
	EXTAL2		—	—	—		When not connec crystal oscillator, Vcc.
	Port L		–0.3	—	AVcc × 0.2		
	Other input pins		–0.3	—	VccQ × 0.2		
Input leak current	All input pins	I Iin I	—	—	1.0	µA	Vin = 0.5 to VccQ
Three-state leak current	I/O, all output pins (off condition)	I Isti I	—	—	1.0	µA	Vin = 0.5 to VccQ
Output high voltage	All output pins	V_{OH}	2.4	—	—	V	VccQ = 3.0 V, I_{OH}
			2.0	—	—		VccQ = 3.0 V, I_{OH}
Output low voltage	All output pins	V_{OL}	—	—	0.55		VccQ = 3.6 V, I_{OL}
Pull-up resistance	Port pin	Rpull	30	60	120	kΩ	
Pin capacity	All pins	C	—	—	10	pF	

	and D/A conversion					
	Idle	—	1	20	µA	Ta = 25°C

Notes: Even when PLL is not used, always connect Vcc-PLL1 and Vcc-PLL2 to Vcc and Vss-PLL1 and Vss-PLL2 to Vss.

Even when RTC is not used, always supply power between Vcc-RTC and Vss-R

AVcc must be under condition of VccQ – 0.3 V ≤ AVcc ≤ VccQ + 0.3 V. If the A/D converters are not used, do not leave the AVcc and AVss pins open. Connect AV VccQ, and connect AVss to VssQ.

Current dissipation values shown are the values at which all output pins are with under conditions of V_{IH} min = VccQ – 0.5 V, V_{IL} max = 0.5 V.

The same voltage should be supplied to Vcc, Vcc-RTC, Vcc-PLL1, and Vcc-PLL

* If the IRL and IRLS interrupts are used, the minimum is 1.9 V.

Table 23.3 Permitted Output Current Values

VccQ = 3.3 ± 0.3 V, Vcc = 1.55 to 2.15 V, AVcc = 3.3 ± 0.3 V, Ta = –20 to 75°C

Item	Symbol	Min	Typ	Max
Output low-level permissible current (per pin)	I_{OL}	—	—	2.0
Output low-level permissible current (total)	$\sum I_{OL}$	—	—	120
Output high-level permissible current (per pin)	$-I_{OH}$	—	—	2.0
Output high-level permissible current (total)	$\sum (-I_{OH})$	—	—	40

Caution: To ensure LSI reliability, do not exceed the value for output current given in tab

Operating frequency	CPU, cache, TLB	f	30	—	200	MHz	200
			25		167		167
					133		133
					100		100
	External bus		30	—	66.67		200
			25				167 133 100
	Peripheral module		7.5	—	33.34		200
			6.25				167 133 100

EXTAL clock input cycle time (clock mode 2)	t_{EXcyc}	60	160	ns	
EXTAL clock input low pulse width	t_{EXL}	1.5	—	ns	
EXTAL clock input high pulse width	t_{EXH}	1.5	—	ns	
EXTAL clock input rise time	t_{EXR}	—	6	ns	
EXTAL clock input fall time	t_{EXF}	—	6	ns	
CKIO clock input frequency	f_{CKI}	20	66	MHz	2
CKIO clock input cycle time	t_{CKIcyc}	15.2	40	ns	
CKIO clock input low pulse width	t_{CKIL}	1.5	—	ns	
CKIO clock input high pulse width	t_{CKIH}	1.5	—	ns	
CKIO clock input rise time	t_{CKIR}	—	6	ns	
CKIO clock input fall time	t_{CKIF}	—	6	ns	
CKIO clock output frequency	f_{OP}	25	66	MHz	2
CKIO clock output cycle time	t_{cyc}	15.2	40	ns	
CKIO clock output low pulse width	t_{CKOL}	3	—	ns	
CKIO clock output high pulse width	t_{CKOH}	3	—	ns	
CKIO clock output rise time	t_{CKOR}	—	5	ns	
CKIO clock output fall time	t_{CKOF}	—	5	ns	
CKIO2 clock output delay time	t_{CK2D}	–3	3	ns	
CKIO2 clock output rise time	t_{CK20R}	—	7	ns	
CKIO2 clock output fall time	t_{CK20F}	—	7	ns	
Power-on oscillation settling time	t_{OSC1}	10	—	ms	2
$\overline{\text{RESETP}}$ setup time	t_{RESPS}	20	—	ns	2
$\overline{\text{RESETM}}$ setup time	t_{RESMS}	6	—	ns	
$\overline{\text{RESETP}}$ assert time	t_{RESPW}	20	—	t_{cyc}	
$\overline{\text{RESETM}}$ assert time	t_{RESMW}	20	—	t_{cyc}	
Standby return oscillation settling time 1	t_{OSC2}	10	—	ms	2
Standby return oscillation settling time 2	t_{OSC3}	10	—	ms	2
Standby return oscillation settling time 3	t_{OSC4}	11	—	ms	2
PLL synchronization settling time 1 (standby canceled)	t_{PLL1}	100	—	µs	2
PLL synchronization settling time 2 (multiplication rete modified)	t_{PLL2}	100	—	µs	2
IRQ/IRL interrupt determination time (RTC used and standby mode)	t_{IRLSTB}	100	—	µs	2

Note: * The clock input from the EXTAL pin.

Figure 23.1 EXTAL Clock Input Timing

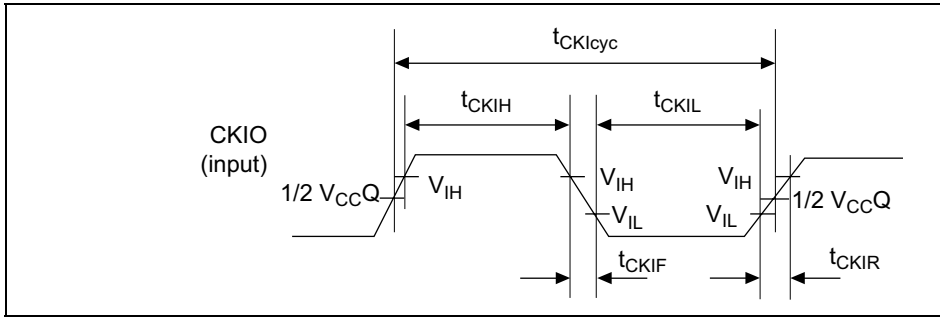


Figure 23.2 CKIO Clock Input Timing

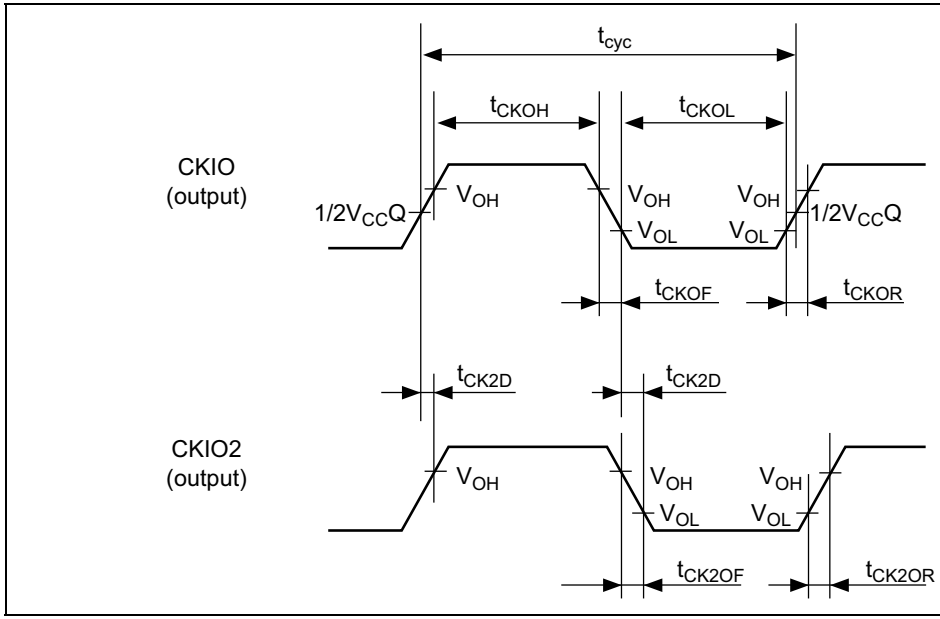
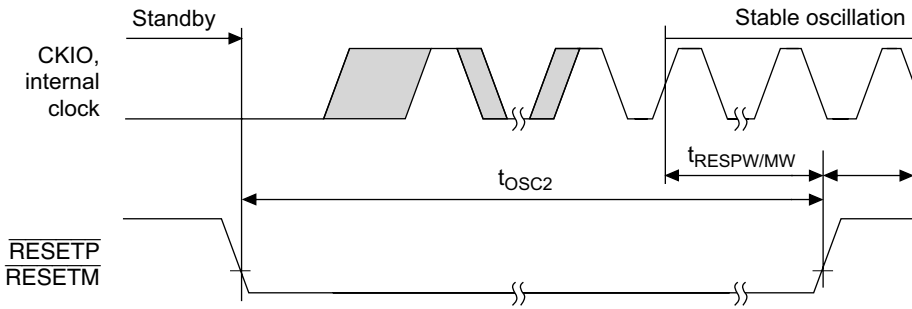


Figure 23.3 CKIO Clock Output Timing

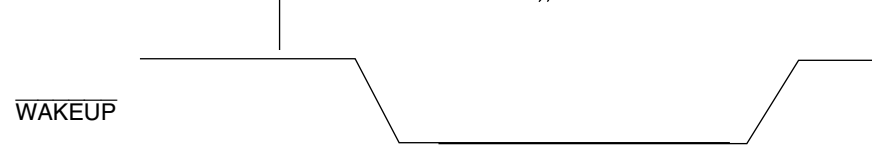
Note: Oscillation settling time when built-in oscillator is used

Figure 23.4 Power-on Oscillation Settling Time



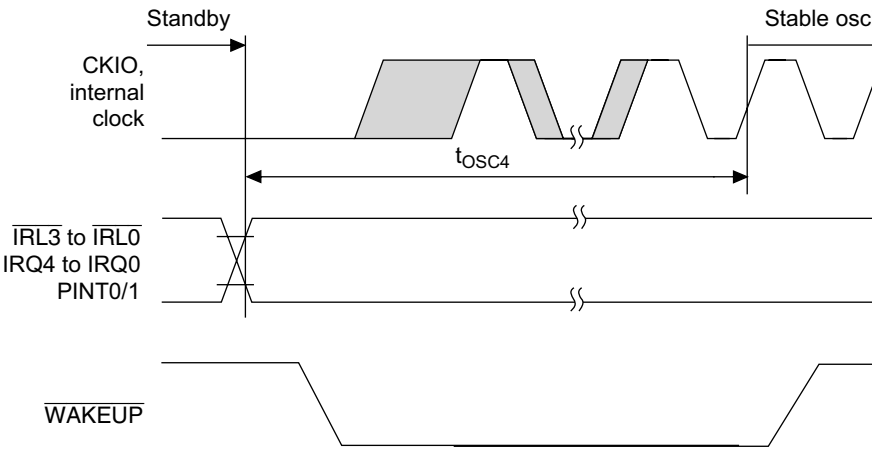
Note: Oscillation settling time when built-in oscillator is used in the oscillation off mode

Figure 23.5 Oscillation Settling Time at Standby Return (Return by Res



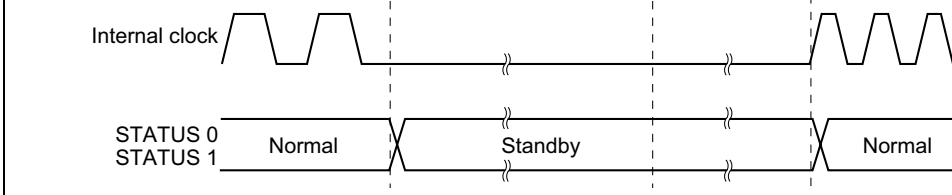
Note: Oscillation settling time when built-in oscillator is used in the oscillation off

Figure 23.6 Oscillation Settling Time at Standby Return (Return by NM



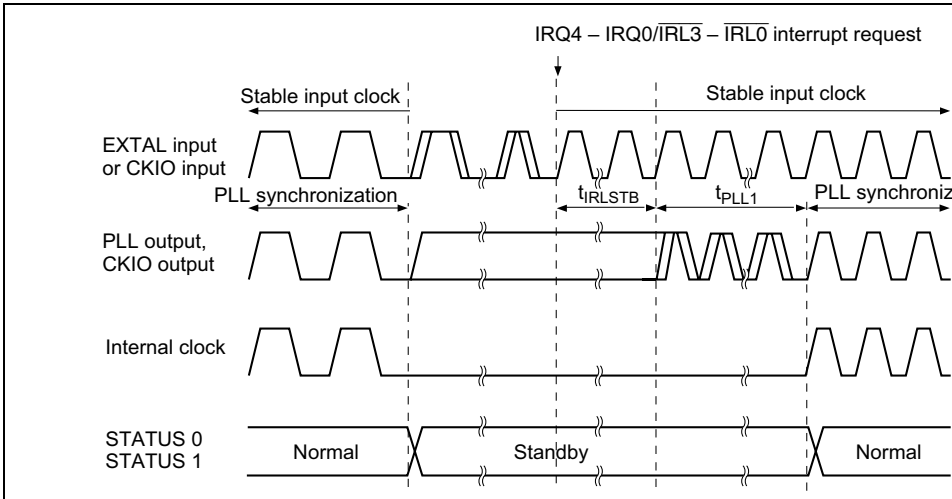
Note: Oscillation settling time when built-in oscillator is used in the oscillation off mode (only when RTC is used)

Figure 23.7 Oscillation Settling Time at Standby Return (Return by IRQ4 to IRQ0, PINT0/1, IRL3 to IRL0)



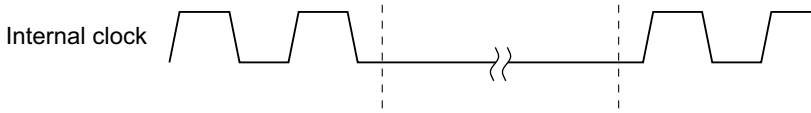
Note: PLL oscillation settling time during the continued oscillation mode or when clock is input from EXTAL pin or CKIO pin

Figure 23.8 PLL Synchronization Settling Time during Standby Recovery (Rese



Note: PLL oscillation settling time during the continued oscillation mode or when clock is input from EXTAL pin or CKIO pin

Figure 23.9 PLL Synchronization Settling Time during Standby Recove (IRQ/IRL or PINT0/PINT1 Interrupt)



Notes: 1. CKIO input in clock mode 7
2. PLL output in other than clock mode 7

Figure 23.10 PLL Synchronization Settling Time when Frequency Multipl
Rate Modified

$\overline{\text{RESETP}}$ hold time	t_{RESPH}	4	—	ns
$\overline{\text{RESETM}}$ pulse width	t_{RESMW}	20 *3	—	tcyc
$\overline{\text{RESETM}}$ setup time	t_{RESMS}	6	—	ns
$\overline{\text{RESETM}}$ hold time	t_{RESMH}	34	—	ns
$\overline{\text{BREQ}}$ setup time	t_{BREQS}	6	—	ns
$\overline{\text{BREQ}}$ hold time	t_{BREQH}	4	—	ns
NMI setup time *1	t_{NMIS}	10	—	ns
NMI hold time	t_{NMIH}	4	—	ns
IRQ5–IRQ0 setup time *1	t_{IRQS}	10	—	ns
IRQ5–IRQ0 hold time	t_{IRQH}	4	—	ns
$\overline{\text{IRQOUT}}$ delay time	t_{IRQOD}	—	10	ns
$\overline{\text{BACK}}$ delay time	t_{BACKD}	—	10	ns
STATUS1, STATUS0 delay time	t_{STD}	—	10	ns
Bus tri-state delay time 1	t_{BOFF1}	0	15	ns
Bus tri-state delay time 2	t_{BOFF2}	0	15	ns
Bus buffer-on time 1	t_{BON1}	0	15	ns
Bus buffer-on time 2	t_{BON2}	0	15	ns

Notes: 1. $\overline{\text{RESETP}}$, NMI, and IRQ5 to IRQ0 are asynchronous. Changes are detected
clock fall when the setup shown is used. When the setup cannot be used, d
can be delayed until the next clock falls.

2. In the standby mode, $t_{RESPW} = t_{OSC1}$ (100 µs) when XTAL oscillation is contin
$t_{RESPW} = t_{OSC2}$ (10 ms) when XTAL oscillation is off. In the sleep mode, t_{RESPW}
(100 µs).
When the clock multiplication ratio is changed, $t_{RESPW} = t_{PLL1}$ (100 µs).

3. In the standby mode, $t_{RESMW} = t_{OSC2}$ (10 ms). In the sleep mode, $\overline{\text{RESETM}}$ m
low until STATUS (0-1) changes to reset (HH). When the clock multiplicatio
changed, $\overline{\text{RESETM}}$ must be kept low until STATUS (0-1) changes to reset (H

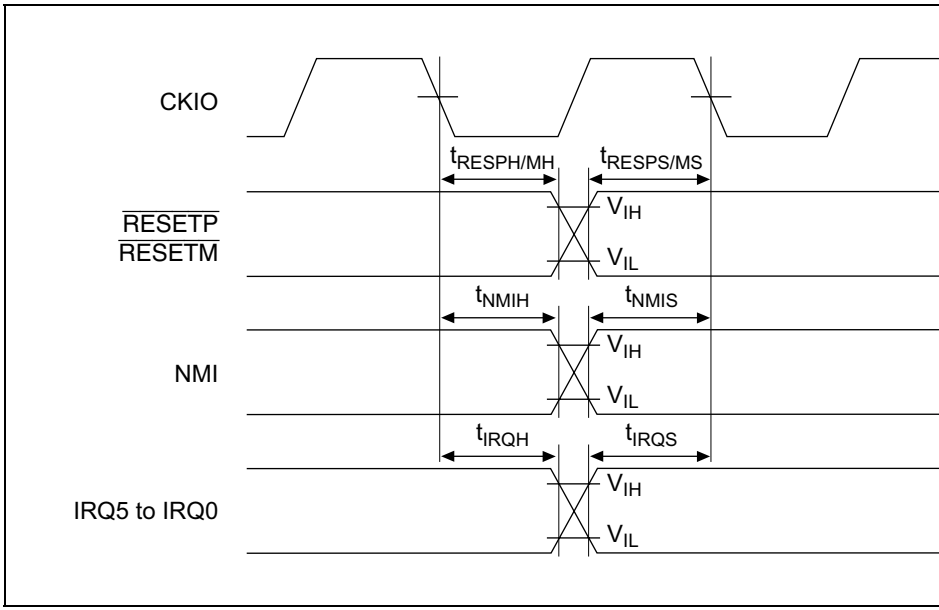


Figure 23.12 Interrupt Signal Input Timing

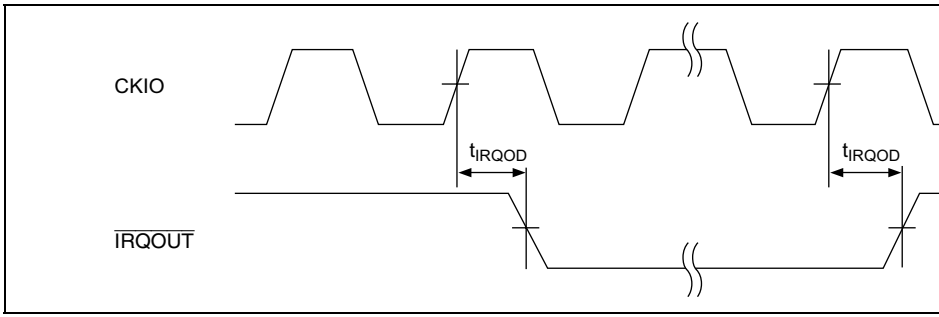


Figure 23.13 $\overline{\text{IRQOUT}}$ Timing

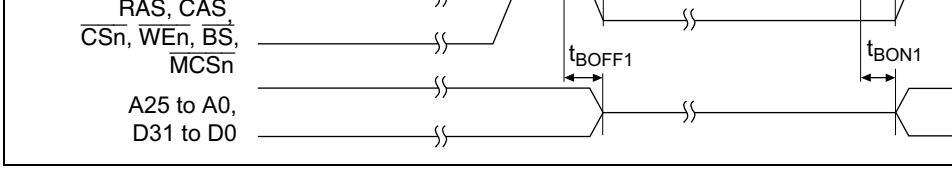


Figure 23.14 Bus Release Timing

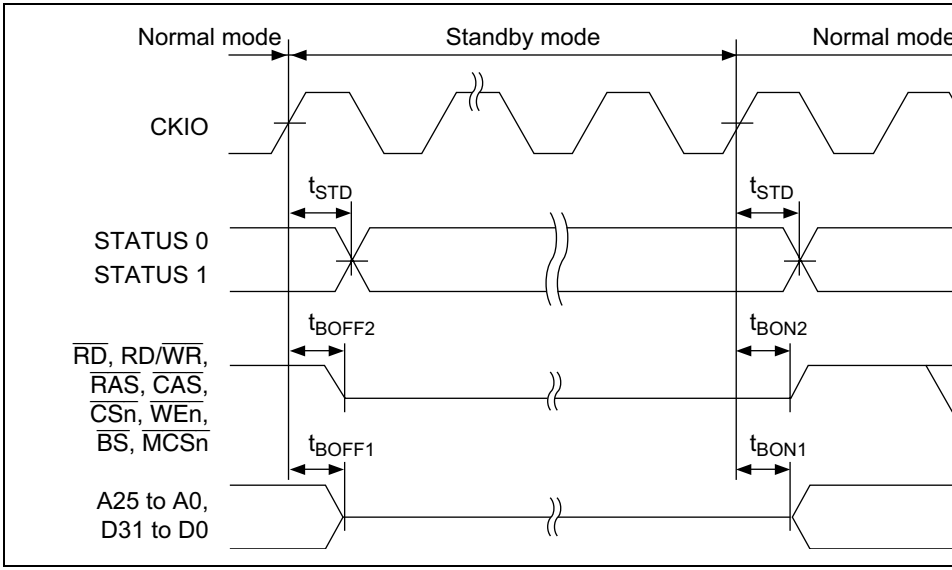


Figure 23.15 Pin Drive Timing at Standby

Address setup time	t_{AS}	0	—	ns	23.16–23.18
Address hold time*1	t_{AH}	4	—	ns	23.16–23.21
$\overline{BS}$ delay time	t_{BSD}	—	10	ns	23.16–23.36, 23.40–23.46
$\overline{CS}$ delay time 1	t_{CSD1}	0	10	ns	23.16–23.21, 23.40–23.46
$\overline{CS}$ delay time 2	t_{CSD2}	—	10	ns	23.16–23.21
$\overline{CS}$ delay time (SDRAM access)	t_{CSD3}	1.5	10	ns	23.22–23.39
Read/write delay time	t_{RWD}	1.5	10	ns	23.16–23.46, 23.39–23.46
Read/write hold time	t_{RWH}	0	—	ns	23.16–23.21
Read strobe delay time	t_{RSD}	—	10	ns	23.16–23.21 23.40–23.43
Read data setup time 1	t_{RDS1}	6	—	ns	23.16–23.21, 23.40–23.46
Read data setup time 2	t_{RDS2}	5	—	ns	23.22–23.25, 23.30–23.33
Read data hold time 1*2	t_{RDH1}	0	—	ns	23.16–23.21, 23.40–23.46
Read data hold time 2	t_{RDH2}	1	—	ns	23.22–23.25, 23.30–23.33
Write enable delay time	t_{WED}	—	10	ns	23.16–23.18, 23.40, 23.41
Write data delay time 1	t_{WDD1}	—	14	ns	23.16–23.18, 23.40, 23.41, 2
Write data delay time 2	t_{WDD2}	1.5	12	ns	23.26–23.29, 23.34–23.36
Write data hold time 1	t_{WDH1}	1.5	—	ns	23.16–23.18, 23.40, 23.41, 2
Write data hold time 2	t_{WDH2}	1.5	—	ns	23.26–23.29, 23.34–23.36
Write data hold time 3	t_{WDH3}	2	—	ns	23.16–23.18
Write data hold time 4	t_{WDH4}	2	—	ns	23.40, 23.41, 23.44–23.46
$\overline{WAIT}$ setup time	t_{WTS}	5	—	ns	23.17–23.21, 23.41, 23.43, 2
$\overline{WAIT}$ hold time	t_{WTH}	0	—	ns	23.17–23.21, 23.41, 23.43, 2
$\overline{RAS}$ delay time 2	t_{RASD2}	1.5	10	ns	23.22–23.39
$\overline{CAS}$ delay time 2	t_{CASD2}	1.5	10	ns	23.22–23.39
DQM delay time	t_{DQMD}	1.5	10	ns	23.22–23.36
CKE delay time	t_{CKED}	1.5	10	ns	23.38

(Reference for CKIO fall)

Notes: 1. Specified based on the slowest negate timing for $\overline{CSn}$, $\overline{RD}$, or $\overline{WEn}$
2. Specified based on whichever negate timing is faster, $\overline{CSn}$ or $\overline{RD}$.

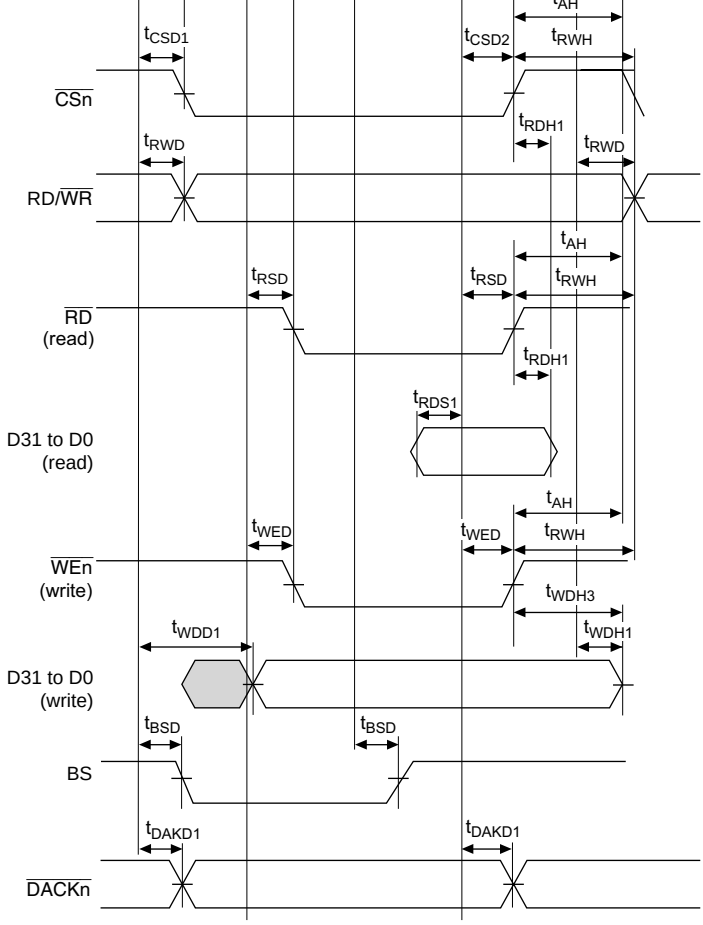


Figure 23.16 Basic Bus Cycle (No Wait)

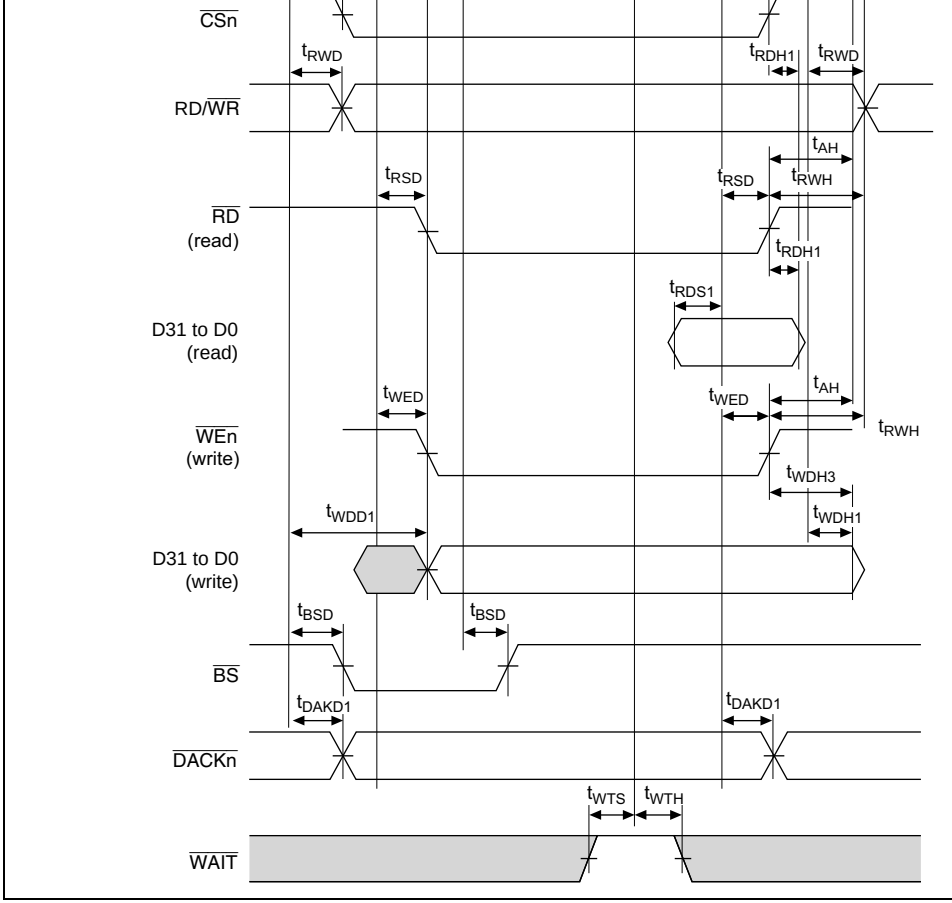


Figure 23.17 Basic Bus Cycle (One Wait)

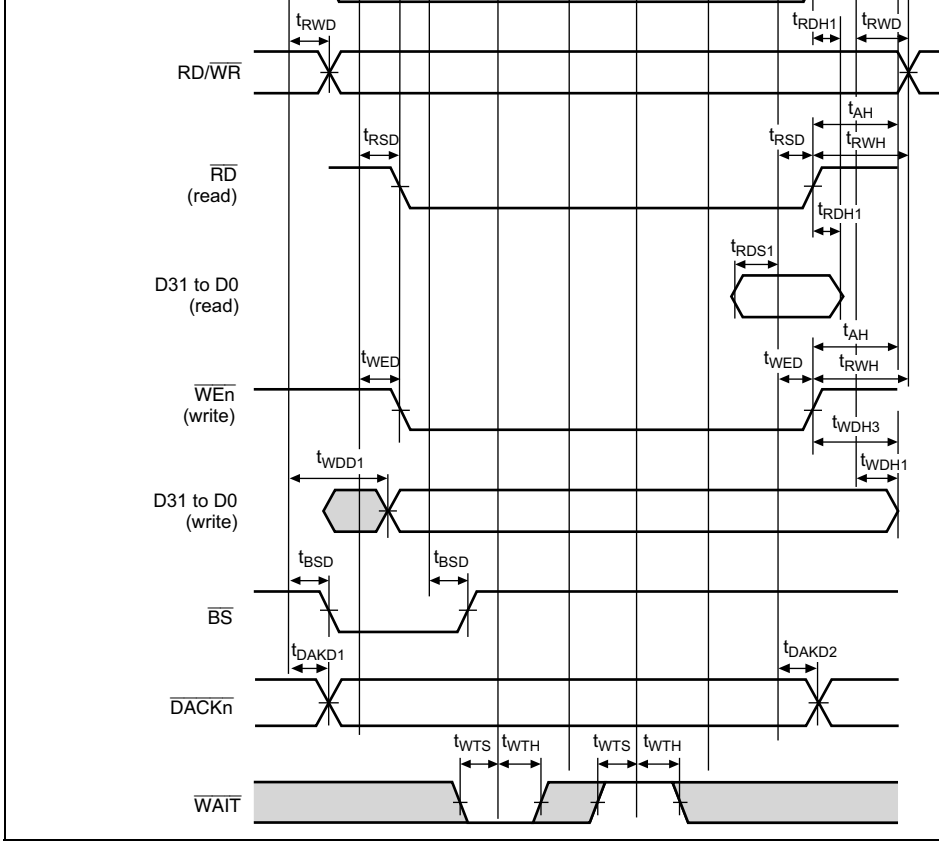
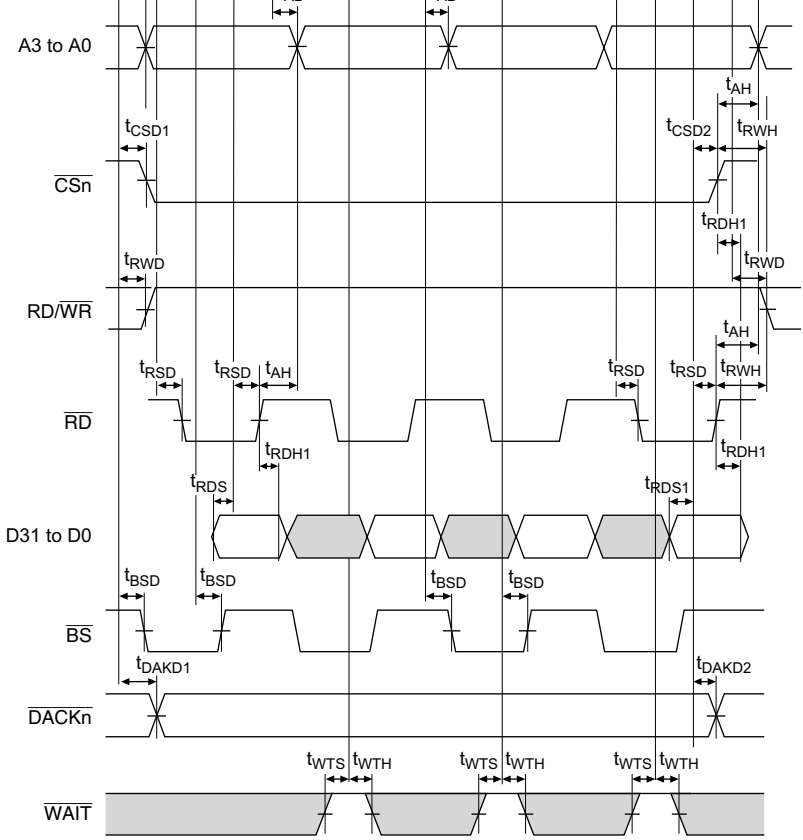
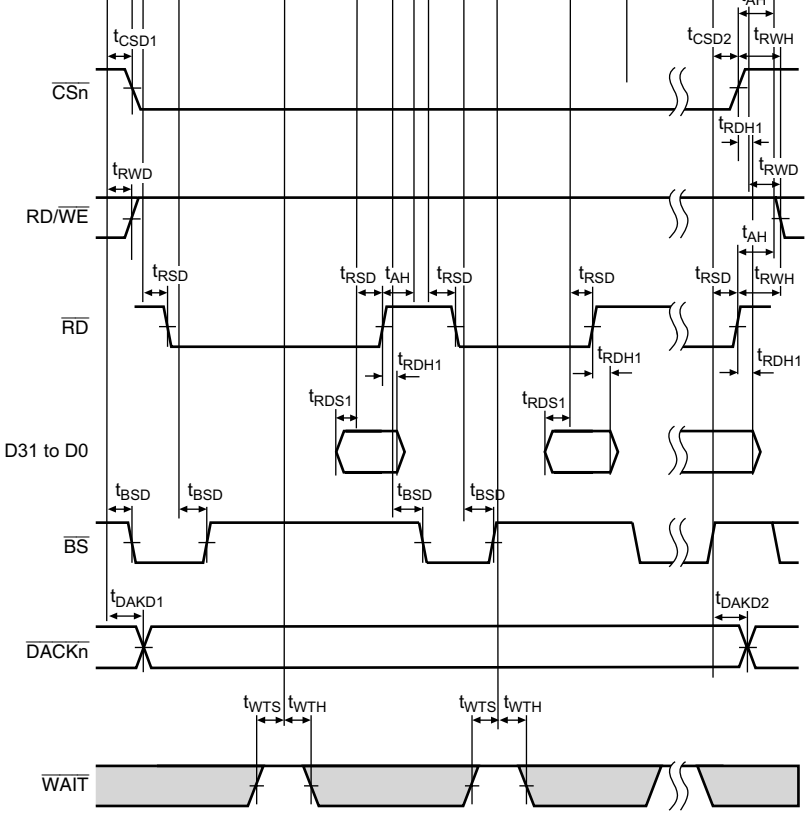


Figure 23.18 Basic Bus Cycle (External Wait, WAITSEL = 1)



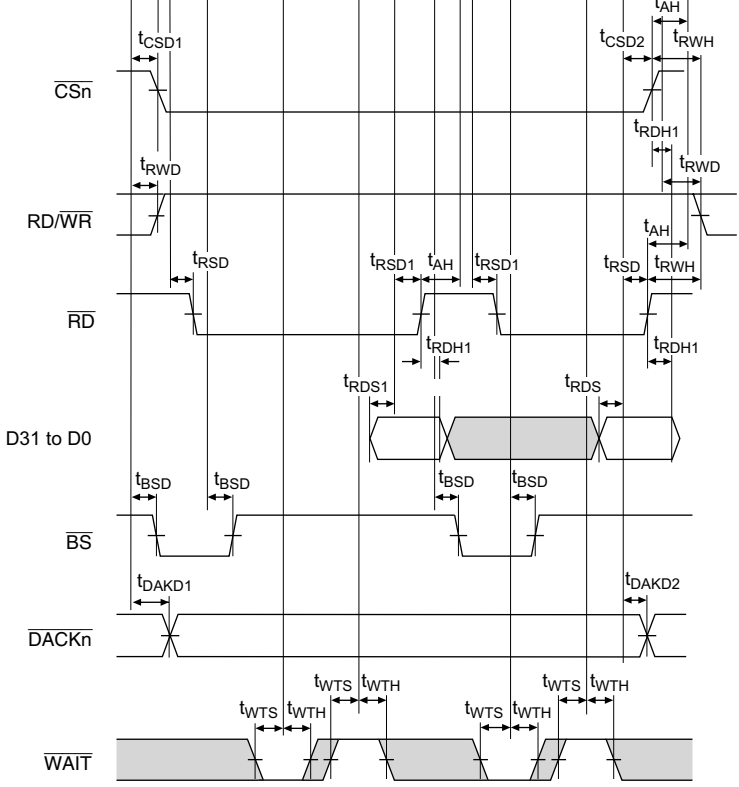
Note: In the write cycle, the basic bus cycle, the basic bus cycle is performed.

Figure 23.19 Burst ROM Bus Cycle (No Wait)



Note: In the write cycle, the basic bus cycle is performed.

Figure 23.20 Burst ROM Bus Cycle (Two Waits)



Note: In the write cycle, the basic bus cycle is performed.

Figure 23.21 Burst ROM Bus Cycle (External Wait, WAITSEL = 1)

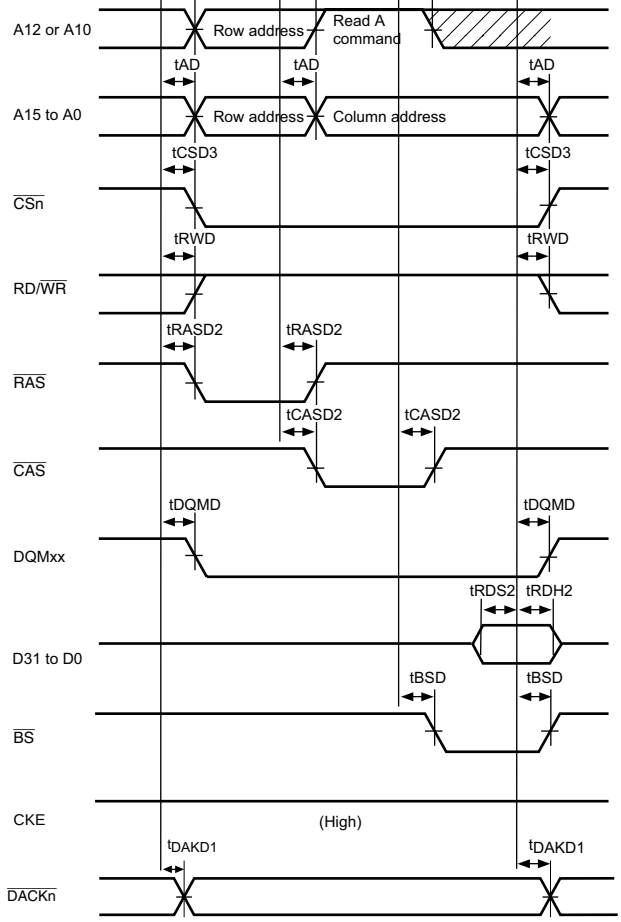


Figure 23.22 Synchronous DRAM Read Bus Cycle (RCD = 0, CAS Latency = 1

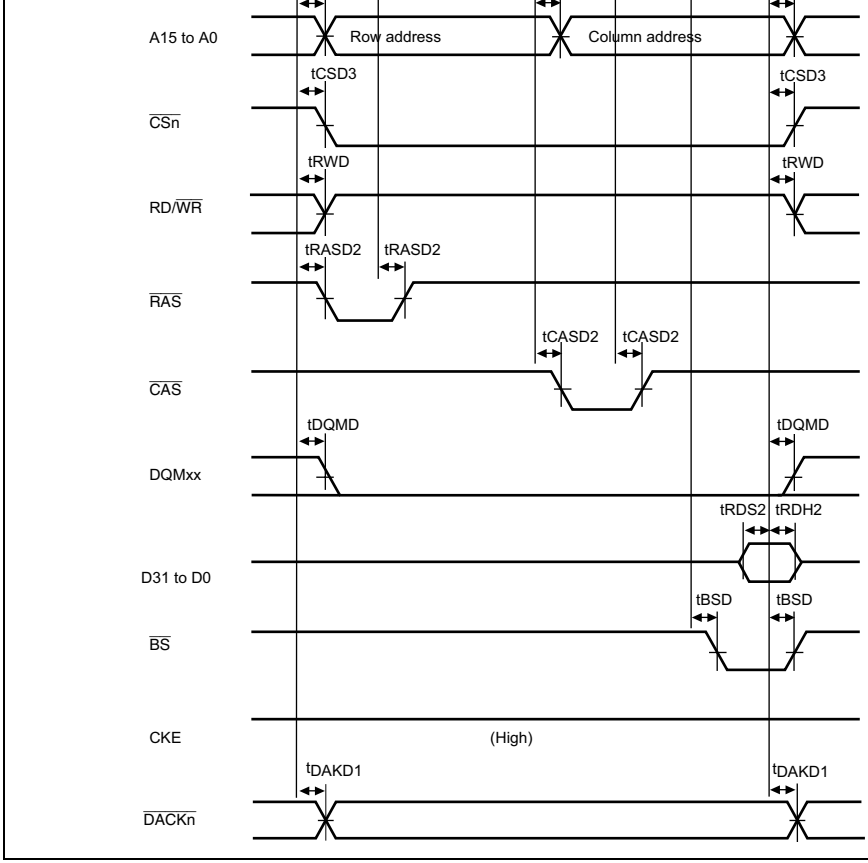


Figure 23.23 Synchronous DRAM Read Bus Cycle (RCD = 2, CAS Latency = 2,

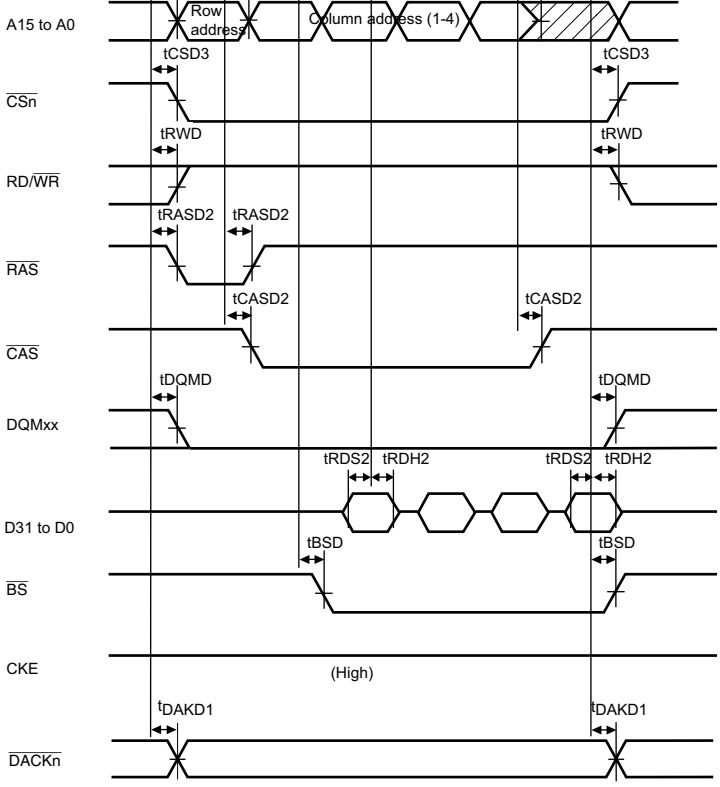


Figure 23.24 Synchronous DRAM Read Bus Cycle (Burst Read (Single Read × 4
CAS Latency = 1, TPC = 1)

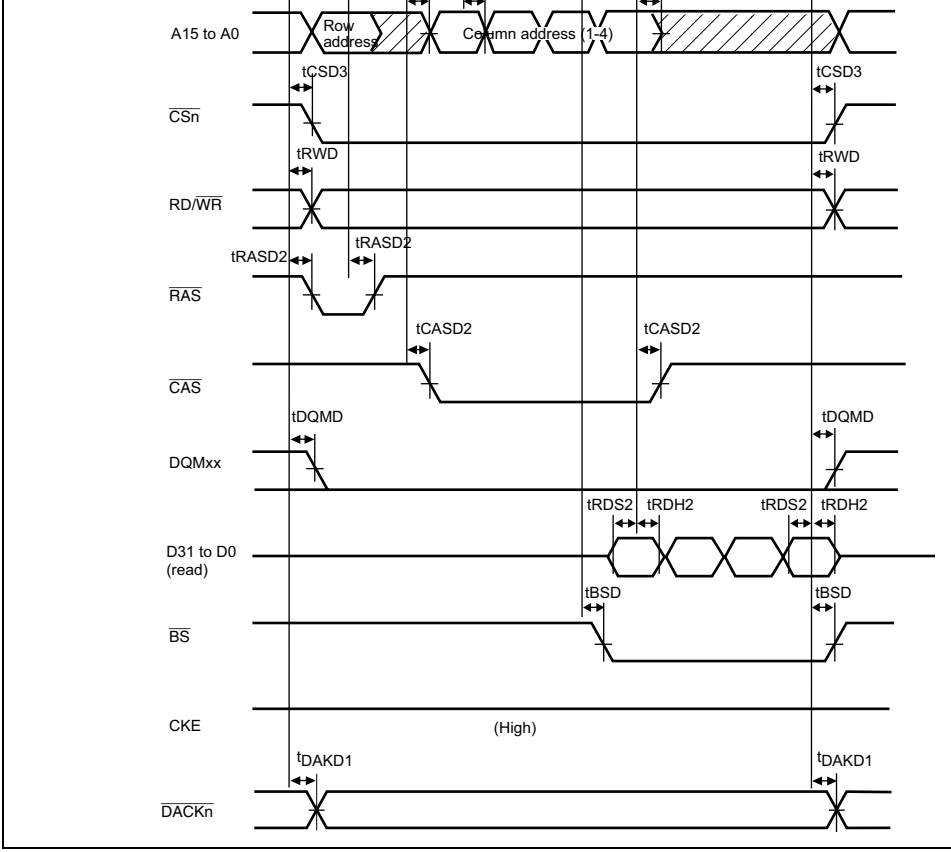


Figure 23.25 Synchronous DRAM Read Bus Cycle (Burst Read (Single Read × 4) CAS Latency = 3, TPC = 0)

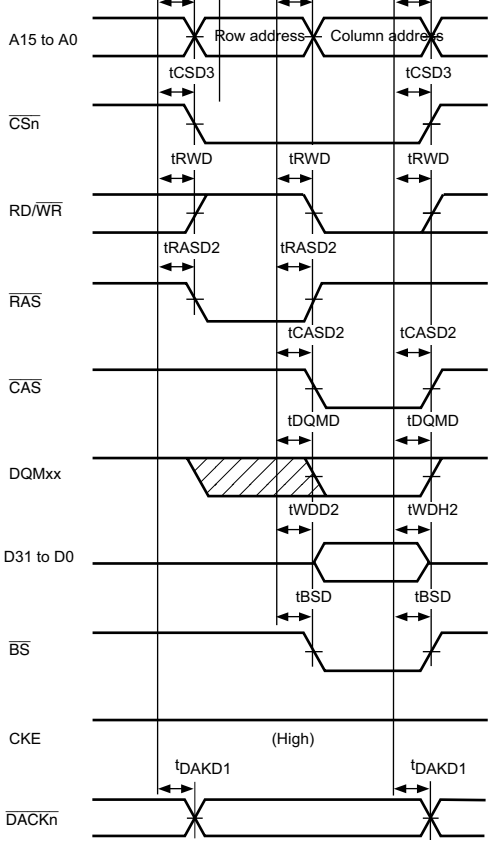


Figure 23.26 Synchronous DRAM Write Bus Cycle (RCD = 0, TPC = 0, TR

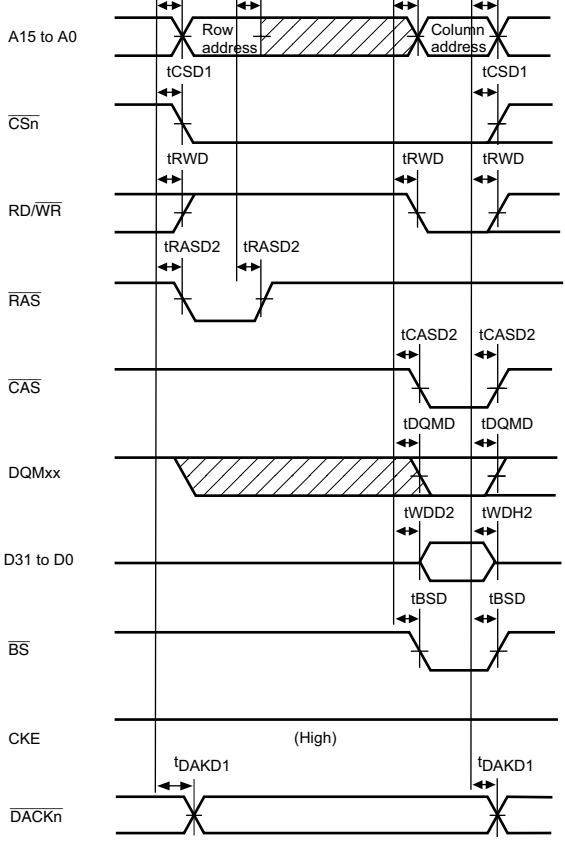


Figure 23.27 Synchronous DRAM Write Bus Cycle (RCD = 2, TPC = 1, TRW

RENESAS

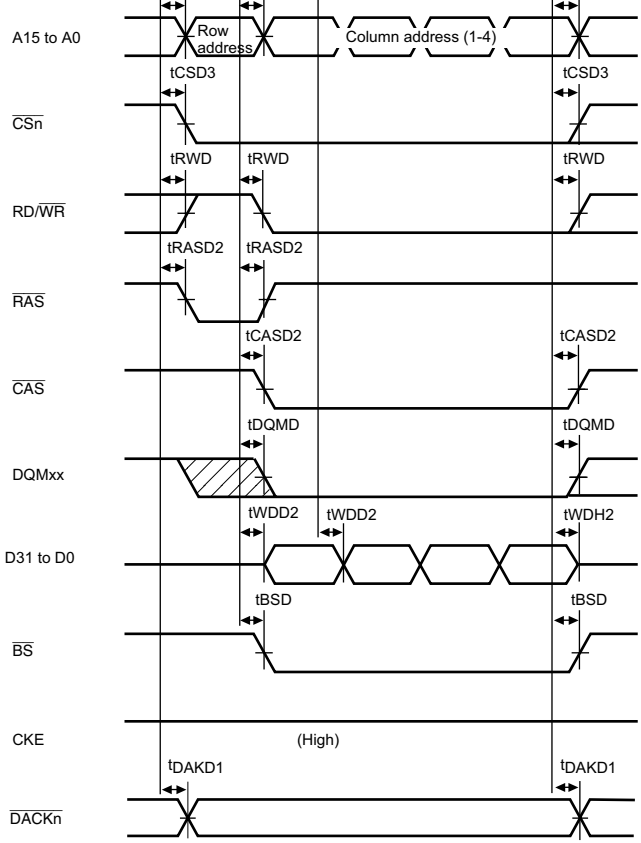


Figure 23.28 Synchronous DRAM Write Bus Cycle (Burst Mode (Single Wr
RCD = 0, TPC = 1, TRWL = 0)

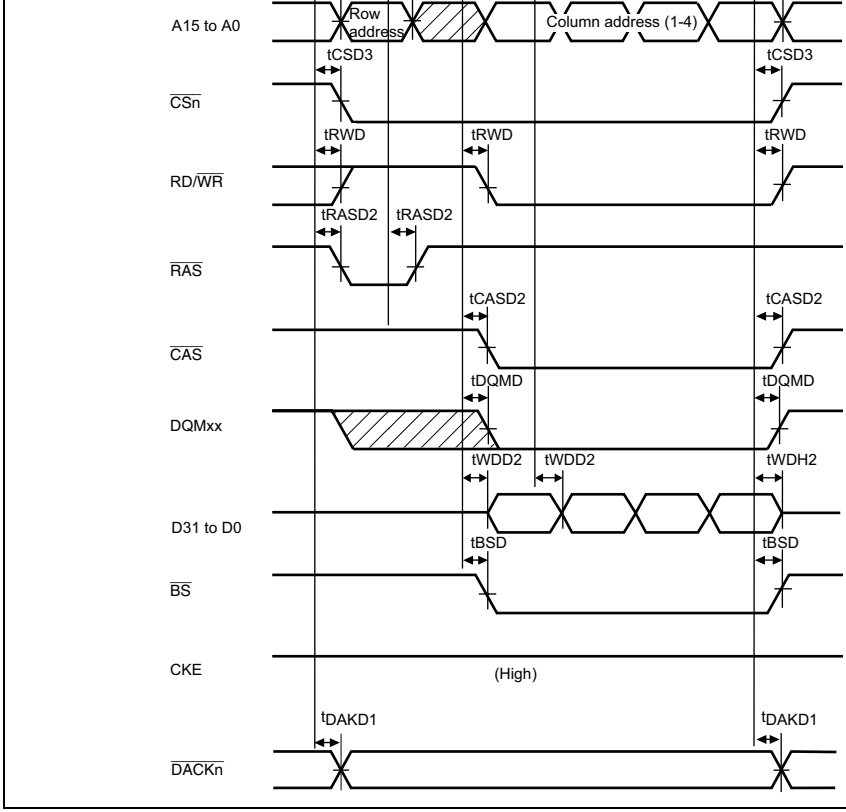


Figure 23.29 Synchronous DRAM Write Bus Cycle (Burst Mode (Single Wri
RCD = 1, TPC = 0, TRWL = 0)

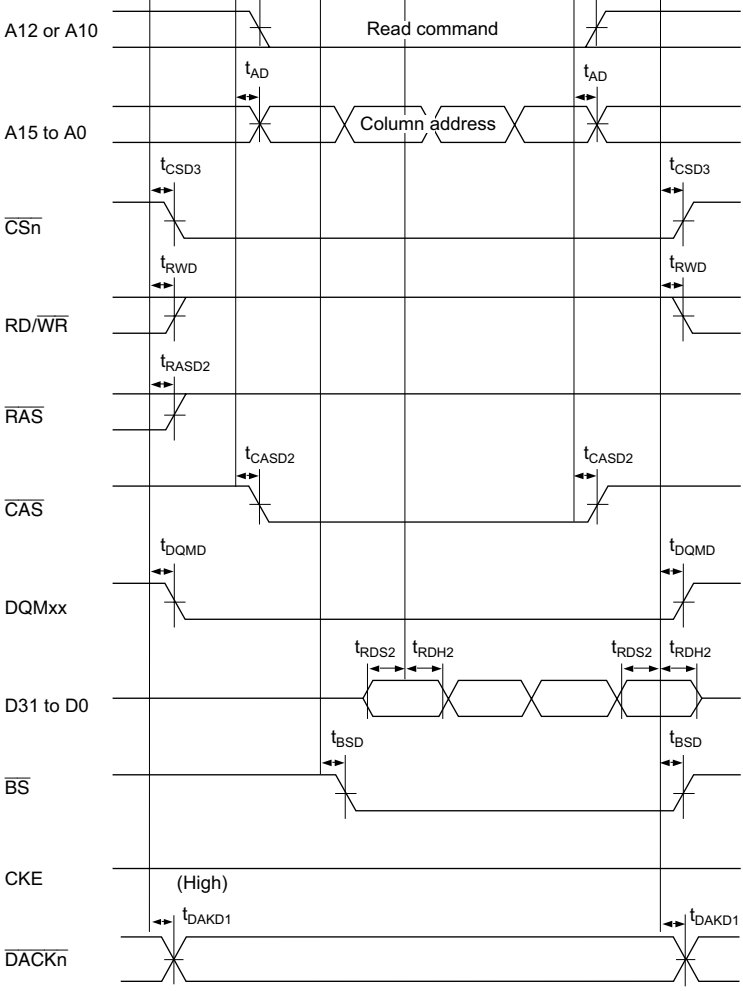


Figure 23.30 Synchronous DRAM Burst Read Bus Cycle (RAS Down, Same Row Address, CAS Latency = 1)

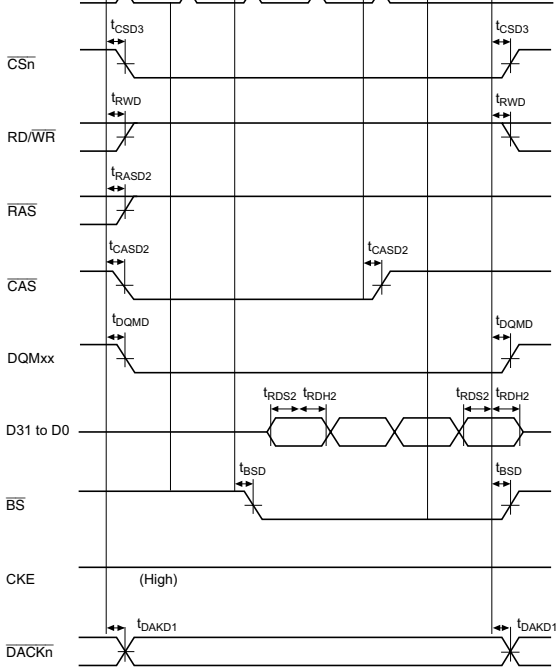


Figure 23.31 Synchronous DRAM Burst Read Bus Cycle (RAS Down, Same Row Address, CAS Latency = 2)

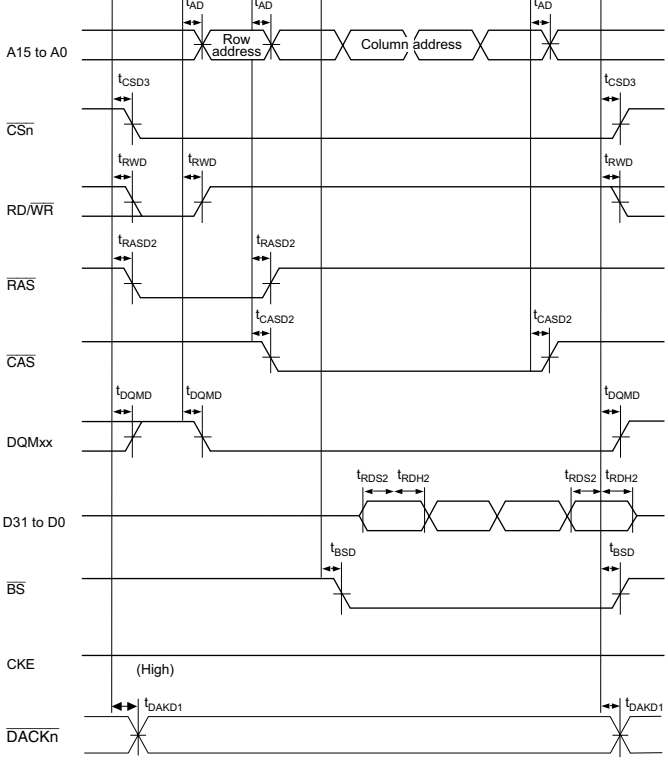


Figure 23.32 Synchronous DRAM Burst Read Bus Cycle (RAS Down, Different Row Address, TPC = 0, RCD = 0, CAS Latency =

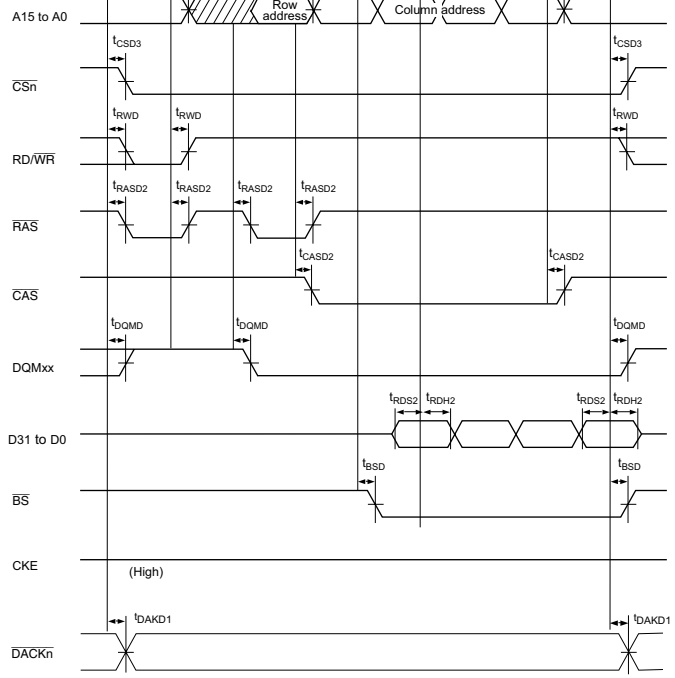


Figure 23.33 Synchronous DRAM Burst Read Bus Cycle (RAS Down, Different Row Address, TPC = 1, RCD = 0, CAS Latency =

RENESAS

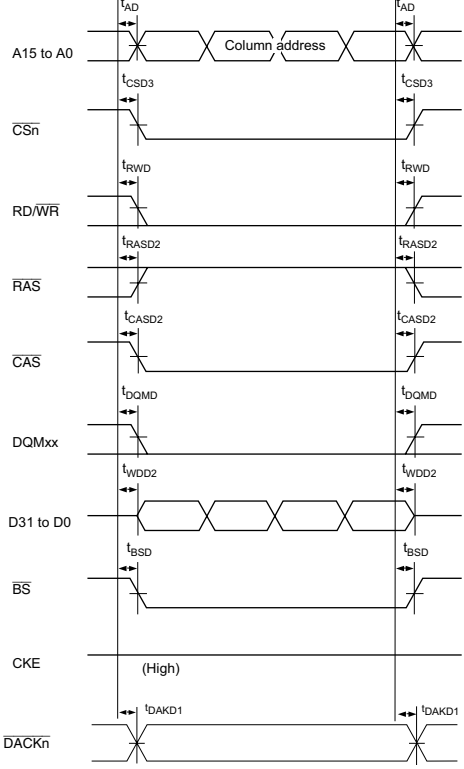


Figure 23.34 Synchronous DRAM Burst Write Bus Cycle (RAS Down, Same Row Address)

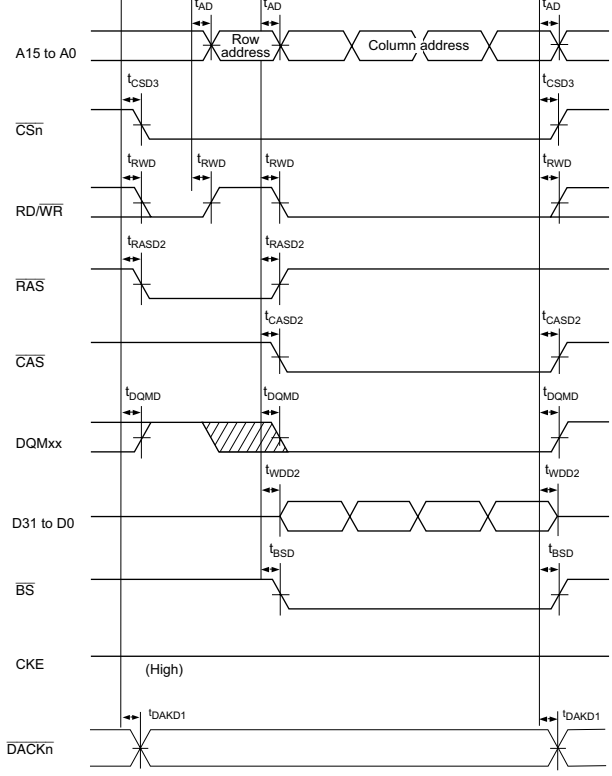


Figure 23.35 Synchronous DRAM Burst Write Bus Cycle (RAS Down, Different Row Address, TPC = 0, RCD = 0)

RENESAS

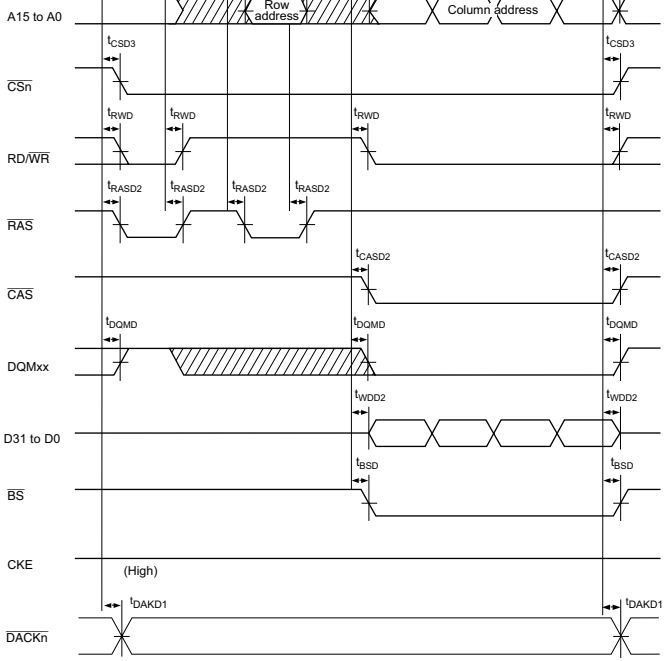


Figure 23.36 Synchronous DRAM Burst Write Bus Cycle (RAS Down, Different Row Address, TPC = 1, RCD = 1)

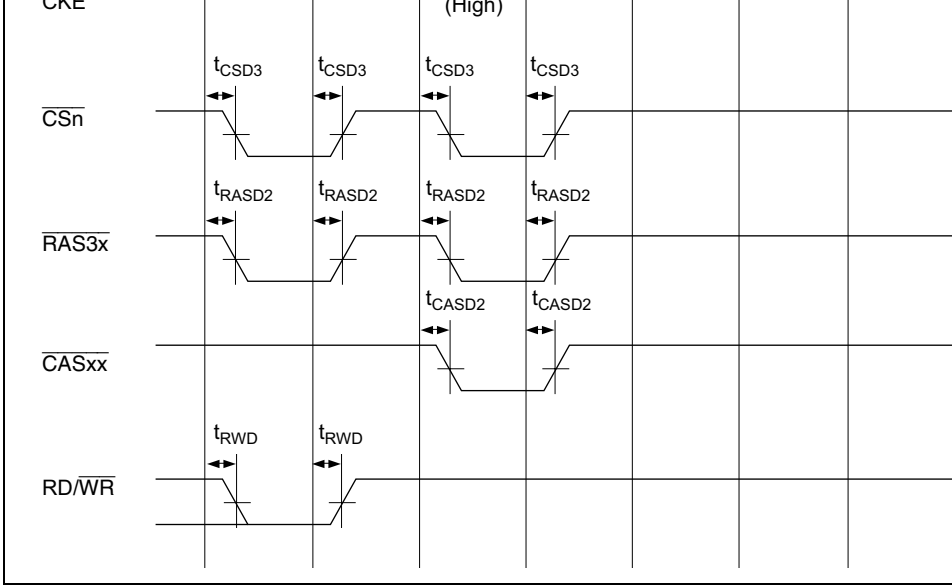


Figure 23.37 Synchronous DRAM Auto-Refresh Timing (TRAS = 1, TPC

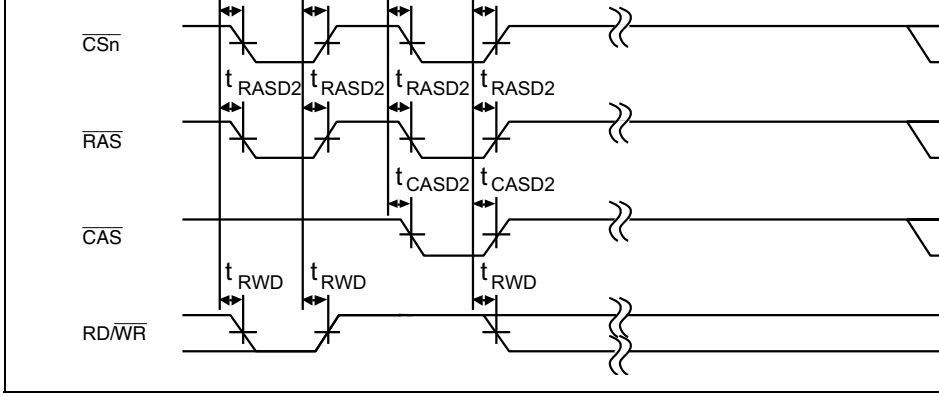


Figure 23.38 Synchronous DRAM Self-Refresh Cycle (TRAS = 1, TPC

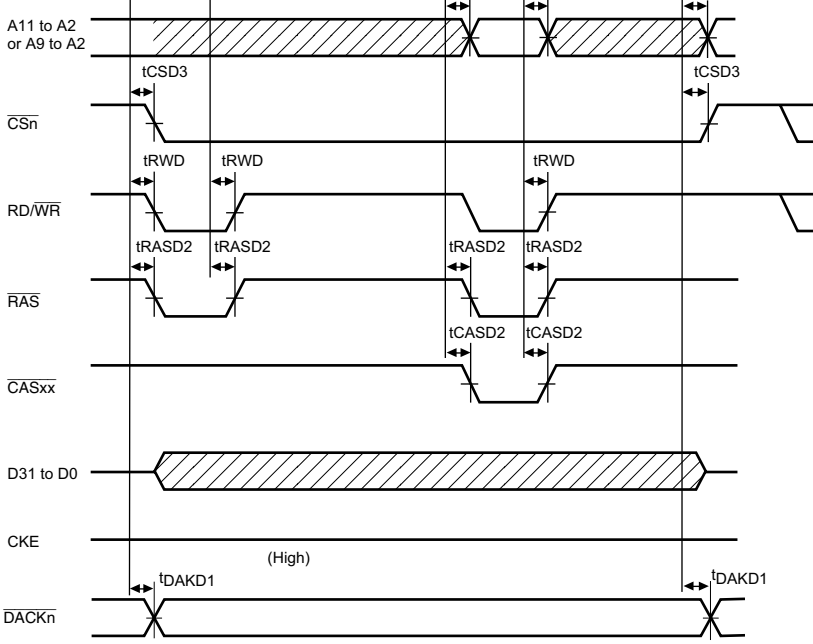


Figure 23.39 Synchronous DRAM Mode Register Write Cycle

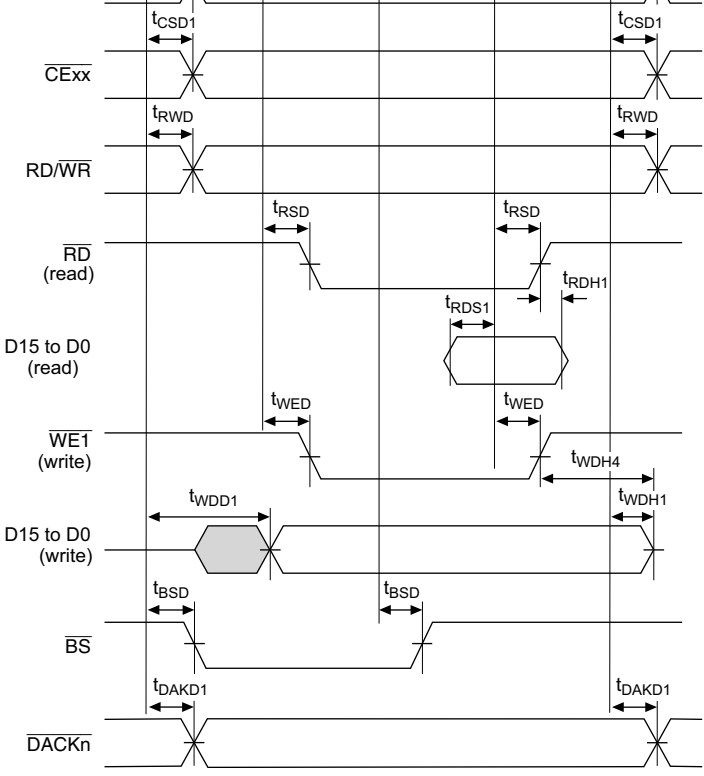


Figure 23.40 PCMCIA Memory Bus Cycle (TED = 0, TEH = 0, No Wa

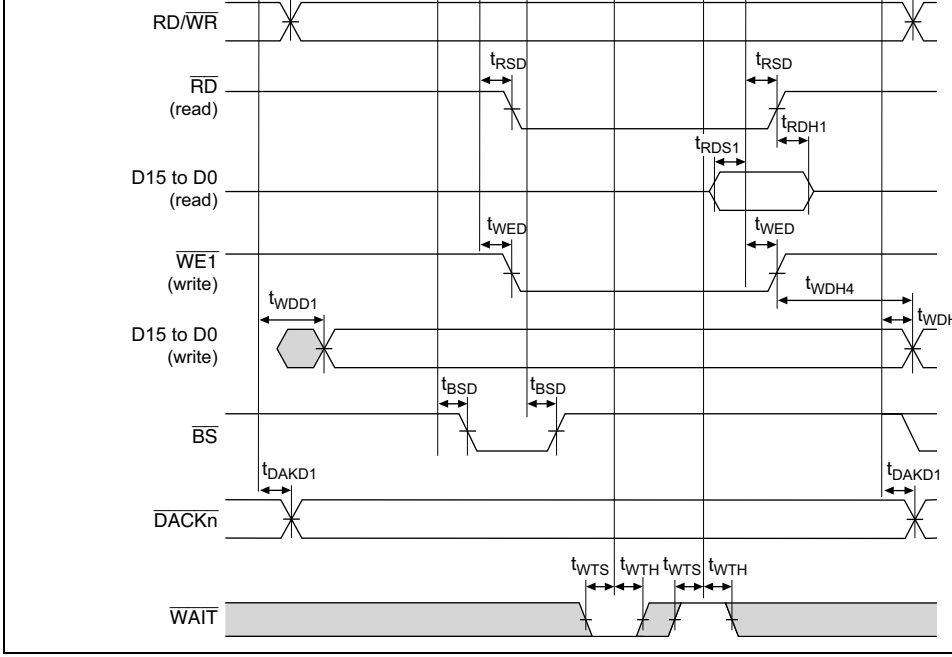
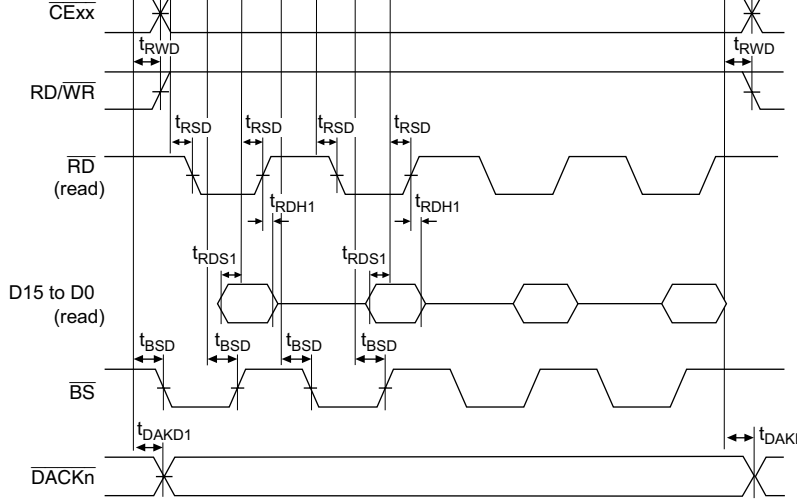
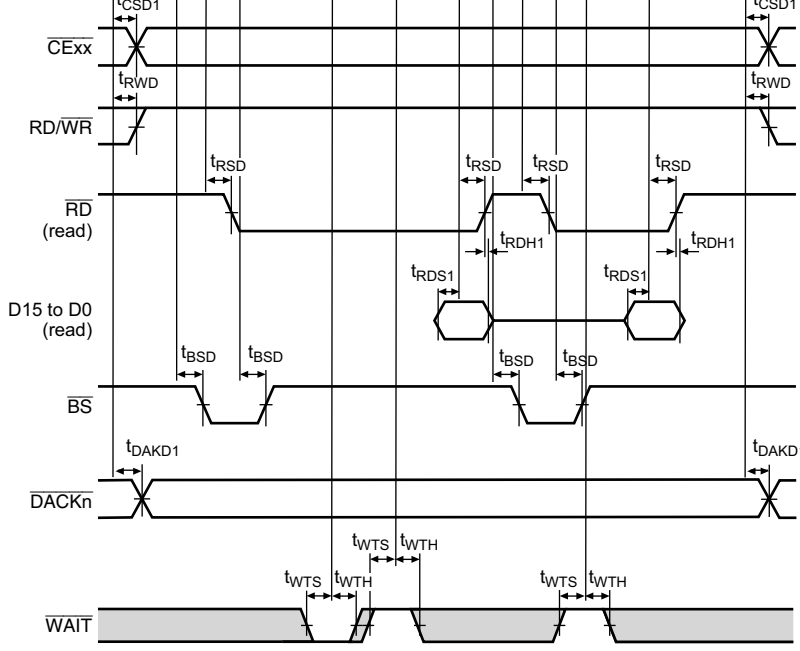


Figure 23.41 PCMCIA Memory Bus Cycle (TED = 2, TEH = 1, One Wait, External Wait, WAITSEL = 1)



Note: Even though burst mode is set, write cycle operation is the same as in normal mode.

Figure 23.42 PCMCIA Memory Bus Cycle (Burst Read, TED = 0, TEH = 0, No Wait)



Note: Even though burst mode is set, the write cycle operation is the same as in normal mode.

Figure 23.43 PCMCIA Memory Bus Cycle
(Burst Read, TED = 1, TEH = 1, Two Waits, Burst Pitch = 3, WAITSEL =

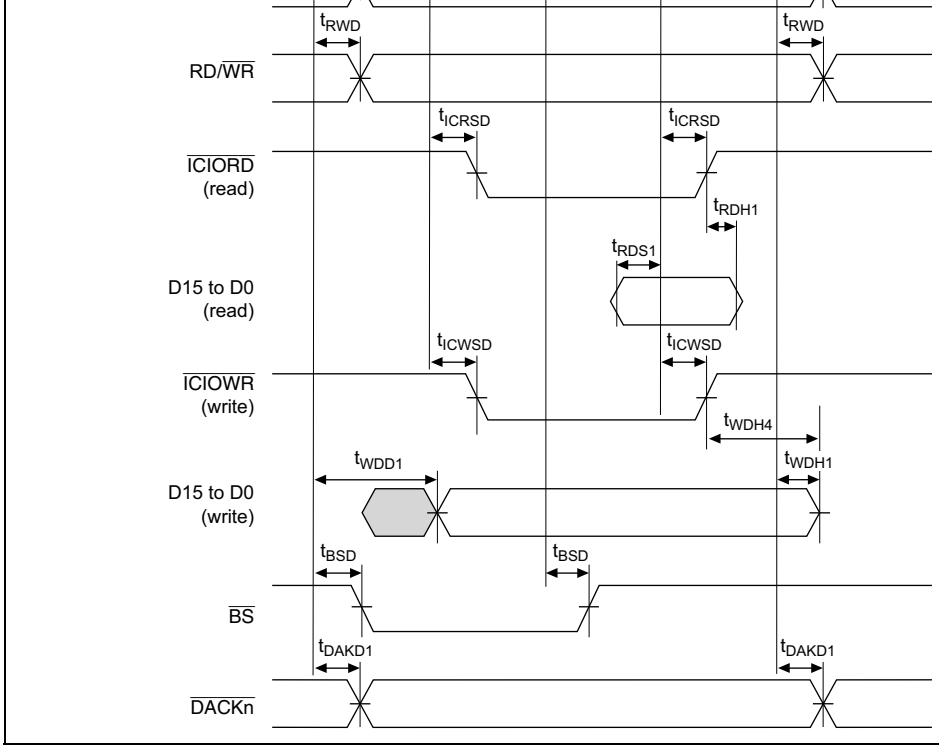


Figure 23.44 PCMCIA I/O Bus Cycle (TED = 0, TEH = 0, No Wait)

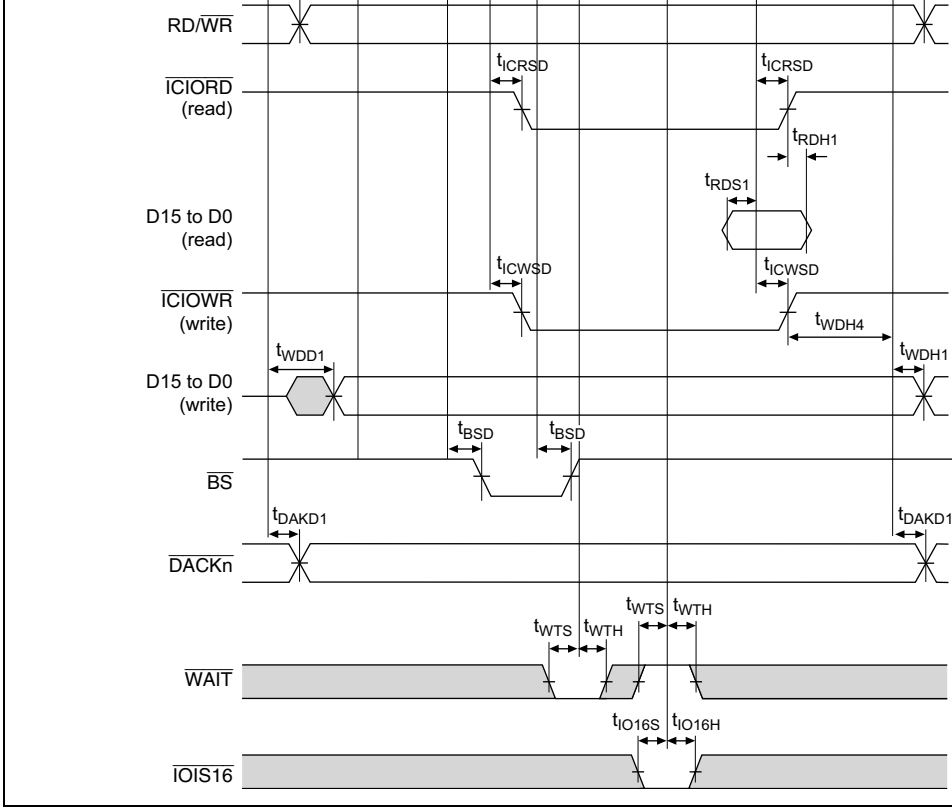


Figure 23.45 PCMCIA I/O Bus Cycle
(TED = 2, TEH = 1, One Wait, External Wait, WAITSEL = 1)

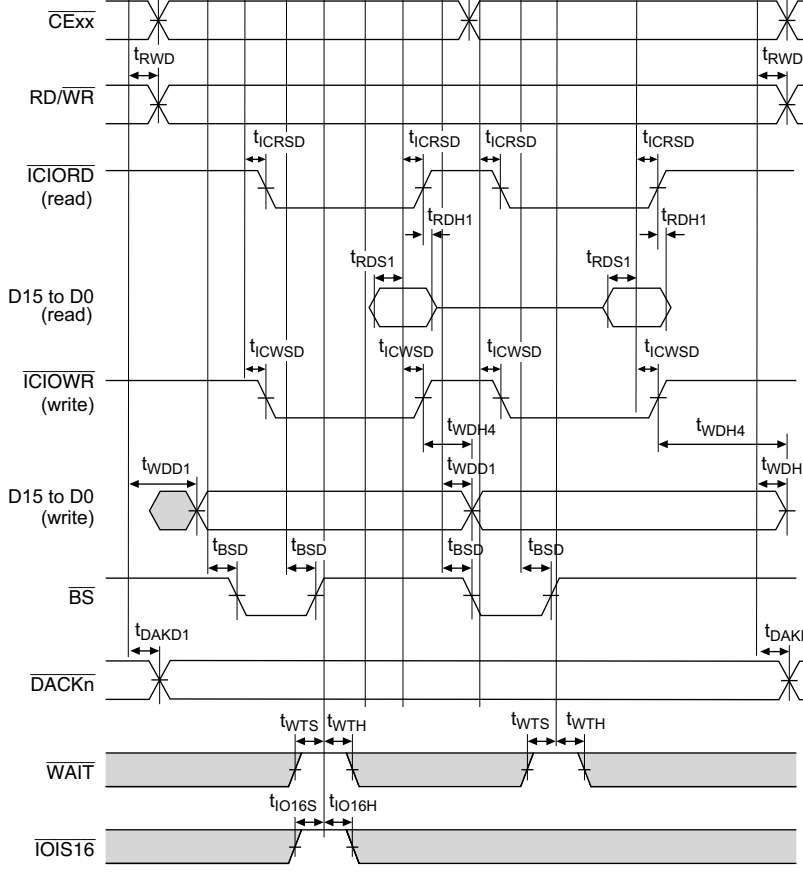


Figure 23.46 PCMCIA I/O Bus Cycle
(TED = 1, TEH = 1, One Wait, Bus Sizing, WAITSEL = 1)

	Timer clock pulse width	Edge specification	t_{TCKWH}	1.5	—	Pcyc
		Both edge specification	t_{TCKWL}	2.5	—	
	Oscillation settling time		t_{ROSC}	3	—	s
SCI	Input clock cycle	Asynchronization	t_{SCYC}	4	—	Pcyc
		Clock synchronization		6	—	
	Input clock rise time		t_{SCKR}	—	1.5	
	Input clock fall time		t_{SCKF}	—	1.5	
	Input clock pulse width		t_{SCKW}	0.4	0.6	tscyc
	Transmission data delay time		t_{TXD}	—	100	ns
	Receive data setup time (clock synchronization)		t_{RXS}	100	—	
	Receive data hold time (clock synchronization)		t_{RXH}	100	—	
	$\overline{\text{RTS}}$ delay time		t_{RTSD}	—	100	
	$\overline{\text{CTS}}$ setup time (clock synchronization)		t_{CTSS}	100	—	
	$\overline{\text{CTS}}$ hold time (clock synchronization)		t_{CTSH}	100	—	
Port	Output data delay time		t_{PORTD}	—	17	ns
	Input data setup time		t_{PORTS1}	15	—	
	Input data hold time		t_{PORTH1}	8	—	
	Input data setup time		t_{PORTS2}	tcyc + 15	—	
	Input data hold time		t_{PORTH2}	8	—	
	Input data setup time		t_{PORTS3}	3 × tcyc + 15	—	
	Input data hold time		t_{PORTH3}	8	—	
DMAC	$\overline{\text{DREQ}}$ setup time		t_{DRQS}	6	—	ns
	$\overline{\text{DREQ}}$ hold time		t_{DREQH}	4	—	
	DRAK delay time		t_{DRAKD}	—	10	

Note: * Pcyc is the P clock cycle.

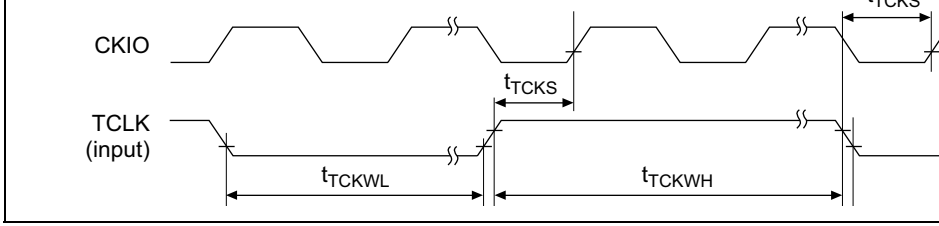


Figure 23.48 TCLK Clock Input Timing

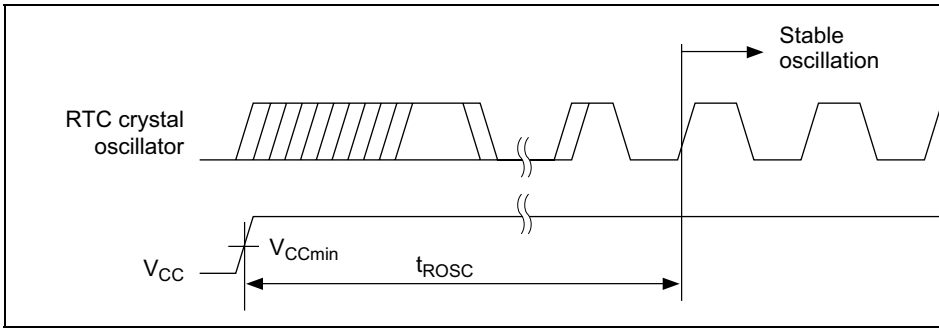


Figure 23.49 Oscillation Settling Time at RTC Crystal Oscillator Power

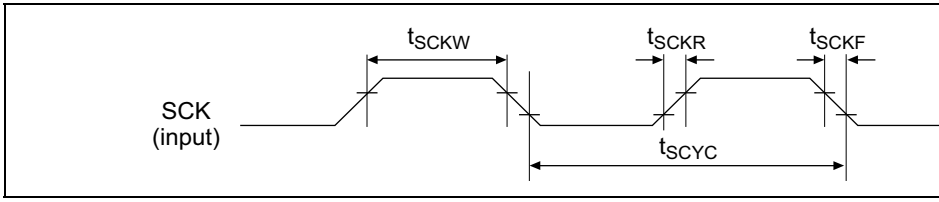


Figure 23.50 SCK Input Clock Timing

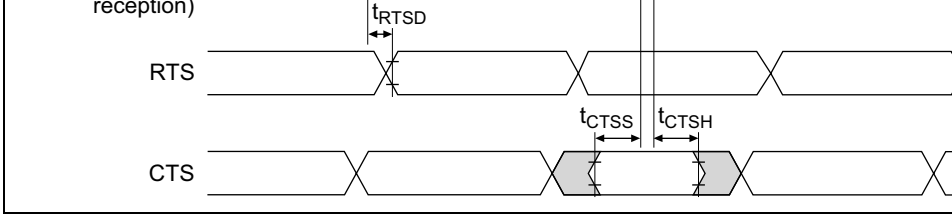


Figure 23.51 SCI I/O Timing in Clock Synchronous Mode

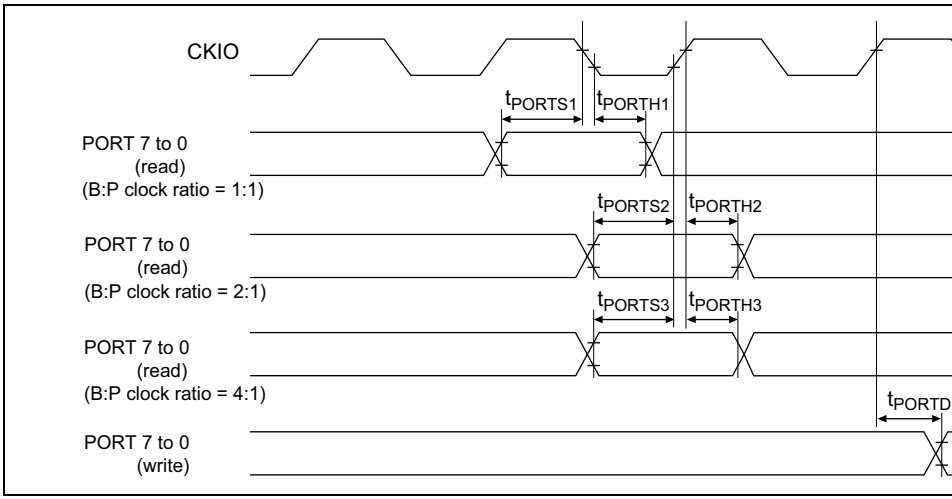


Figure 23.52 I/O Port Timing

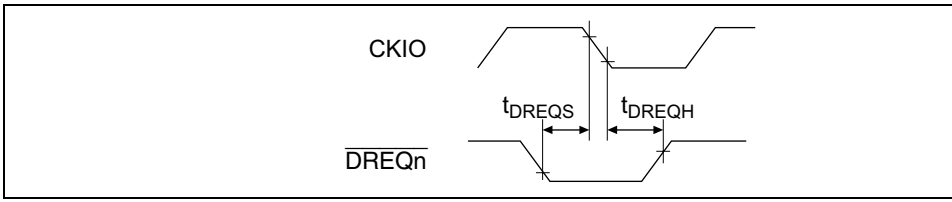
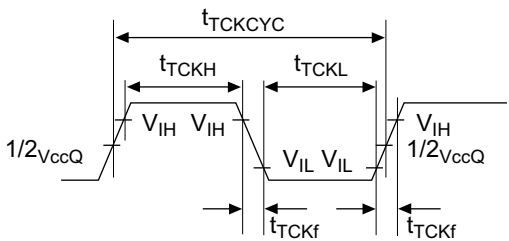


Figure 23.53 $\overline{DREQ}$ Input Timing

Table 23.9 UDI-Related Pin Timing

VccQ = 3.3 ± 0.3V, Vcc = 1.55 to 2.15 V, AVcc = 3.3 ± 0.3V, Ta = –20 to 75°C

Item	Symbol	Min	Max	Unit	Figu
TCK cycle time	t_{TCKCYC}	50	—	ns	23.5
TCK high pulse width	t_{TCKH}	12	—	ns	
TCK low pulse width	t_{TCKL}	12	—	ns	
TCK rise/fall time	t_{TCKf}	—	4	ns	
$\overline{\text{TRST}}$ setup time	t_{TRSTS}	12	—	ns	23.5
$\overline{\text{TRST}}$ hold time	t_{TRSTH}	50	—	t_{cyc}	
TDI setup time	t_{TDIS}	10	—	ns	23.5
TDI hold time	t_{TDIH}	10	—	ns	
TMS setup time	t_{TMSS}	10	—	ns	
TMS hold time	t_{TMSH}	10	—	ns	
TDO delay time	t_{TDOD}	—	16	ns	
$\overline{\text{ASEMD0}}$ setup time	t_{ASEMDH}	12	—	ns	23.5
$\overline{\text{ASEMD0}}$ hold time	t_{ASEMDS}	12	—	ns	



Note: When clock is input from TCK pin

Figure 23.55 TCK Input Timing

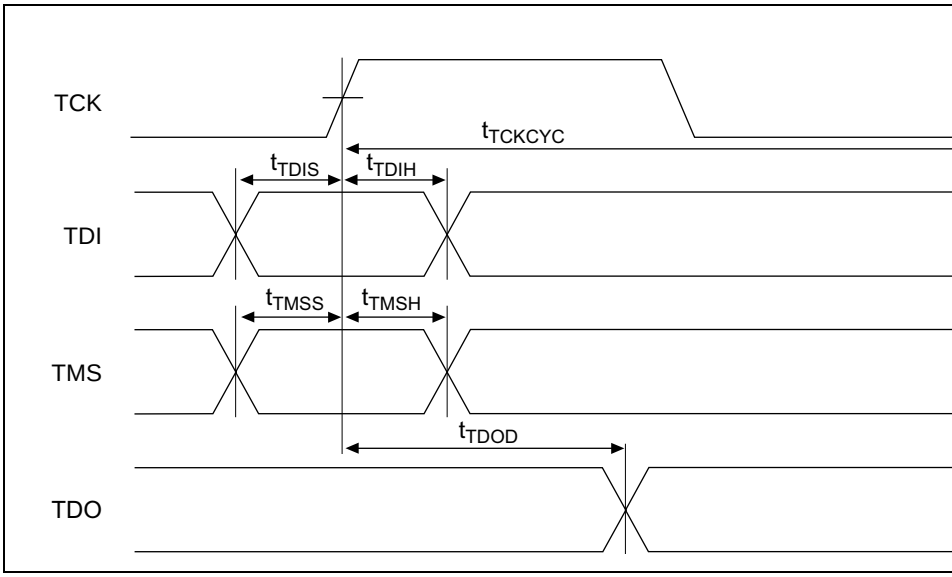


Figure 23.57 UDI Data Transfer Timing

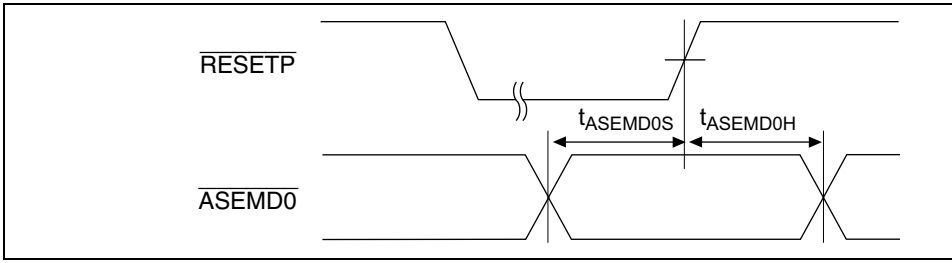
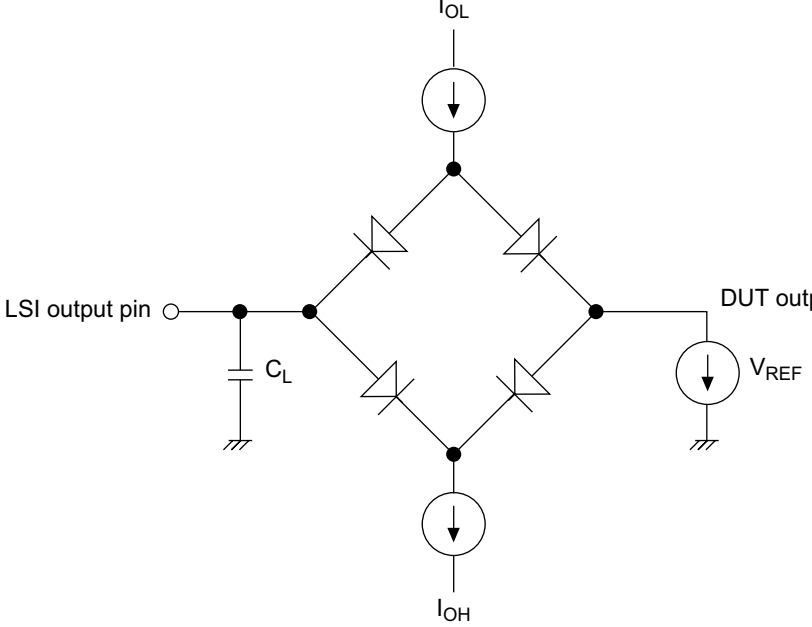


Figure 23.58 $\overline{\text{ASEMD0}}$ Input Timing



Notes: 1. C_L is the total value that includes the capacitance of measureme instruments, etc., and is set as follows for each pin.
30pF: CKIO, $\overline{RAS}$, $\overline{CAS}$, $\overline{CS0}$, $\overline{CS2}$–$\overline{CS6}$, $\overline{CE2A}$, $\overline{CE2B}$, $\overline{BACK}$
50pF: All other pins
2. I_{OL} and I_{OH} are the values shown in table 23.3.

Figure 23.59 Output Load Circuit

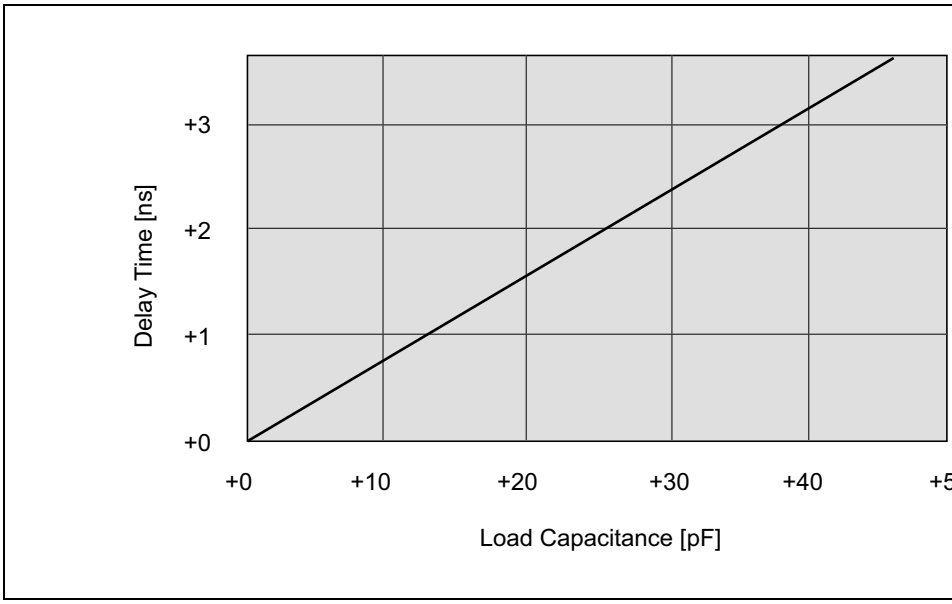


Figure 23.60 Load Capacitance vs. Delay Time

Conversion time	15	—	—	µs
Analog input capacitance	—	—	20	pF
Permissible signal-source (single-source) impedance	—	—	5	kΩ
Nonlinearity error	—	—	±3.0	LS
Offset error	—	—	±2.0	LS
Full-scale error	—	—	±2.0	LS
Quantization error	—	—	±0.5	LS
Absolute accuracy	—	—	±4.0	LS

23.5 D/A Converter Characteristics

Table 23.11 lists the D/A converter characteristics.

Table 23.11 D/A Converter Characteristics

VccQ = 3.3 ± 0.3 V, Vcc = 1.55 to 2.15 V, AVcc = 3.3 ± 0.3 V, Ta = –20 to 75°C

Item	Min	Typ	Max	Unit	Test C
Resolution	8	8	8	bits	
Conversion time	—	—	10.0	µs	20-pF c load
Absolute accuracy	—	±2.5	±4.0	LSB	2-MΩ r load

Category	Pin	Power-On Reset	Manual Reset	Standby	Sleep
Clock	EXTAL	I	I	I	I
	XTAL	O*1	O*1	O*1	O*1
	CKIO	IO*1	IO*1	IO*1	IO*1
	EXTAL2	I	I	I	I
	XTAL2	O	O	O	O
	CAP1, CAP2	—	—	—	—
System control	$\overline{\text{RESETP}}$	I	I	I	I
	$\overline{\text{RESETM}}$	I	I	I	I
	$\overline{\text{BREQ}}$	I	I	I	I
	$\overline{\text{BACK}}$	O	O	O	O
	MD[5:0]	I	I	I	I
	CA	I	I	I	I
	STATUS[1:0]/PTJ[7:6]	O	OP*2	OP*2	OP*2
Interrupt	IRQ[3:0]/$\overline{\text{IRL}}$[3:0]/ PTH[3:0]	V*7	I	I	I
	IRQ[4]/ PTH[4]	V*7	I	I	I
	NMI	I	I	I	I
	$\overline{\text{IRLS}}$[3:0]/PTF[3:0]/ PINT[11:8]	V	I	IZ	I
	$\overline{\text{MCS}}$[7:0]/PTC[7:0]/ PINT[7:0]	V	OP*2	ZH*10 K*2	OP*2
	TCK/PTF[4]/PINT[12]	IV	I	IZ	I
	TDI/PTF[5]/PINT[13]	IV	I	IZ	I
	TMS/PTF[6]/PINT[14]	IV	I	IZ	I
	$\overline{\text{TRST}}$/PTF[7]/PINT[15]	IV	I	IZ	I
	$\overline{\text{IRQOUT}}$	O	O	O	O

	$\overline{CS}$[2:4]/PTK[0:2]	H	OP*2	ZH*10 K*2	OP*2
	$\overline{CS5}$/$\overline{CE1A}$/PTK[3]	H	OP*2	ZH*10 K*2	OP*2
	$\overline{CS6}$/$\overline{CE1B}$	H	O	ZH*10	O
	$\overline{BS}$/PTK[4]	H	OP*2	ZH*10 K*2	OP*2
	$\overline{RAS3L}$/PTJ[0]	H	OP*2	ZOK*3	OP*2
	$\overline{RAS3U}$/PTE[2]	V	OP*2	ZOK*3	OP*2
	$\overline{CASL}$/PTJ[2]	H	OP*2	ZOK*3	OP*2
	$\overline{CASU}$/PTJ[3]	H	OP*2	ZOK*3	OP*2
	$\overline{WE0}$/DQMLL	H	O	ZH*10	O
	$\overline{WE1}$/DQMLU/$\overline{WE}$	H	O	ZH*10	O
	$\overline{WE2}$/DQMUL/$\overline{ICIORD}$/ PTK[6]	H	OP*2	ZH*10 K*2	OP*2
	$\overline{WE3}$/DQMUU/$\overline{ICIOWR}$/ PTK[7]	H	OP*2	ZH*10 K*2	OP*2
	RD/$\overline{WR}$	H	O	ZH*10	O
	$\overline{RD}$	H	O	ZH*10	O
	CKE/PTK[5]	H	OP*2	OK*2	OP*2
	$\overline{WAIT}$	Z	I	Z	I
DMAC	$\overline{DREQ0}$/PTD[4]	V	ZI*6	Z	I
	$\overline{DACK0}$/PTD[5]	V	OP*2	ZK*2	OP*2
	DRAK0/PTD[1]	V	OP*2	ZH*10 K*2	OP*2
	$\overline{DREQ1}$/PTD[6]	V	ZI*6	Z	I
	$\overline{DACK1}$/PTD[7]	V	OP*2	ZK*2	OP*2
	DRAK1/PTD[0]	V	OP*2	ZH*10 K*2	OP*2
Timer	TCLK/PTH[7]	V	ZP	IOP*4	IOP*4

	SCK1/SCPT[3]	V	ZP*2	ZK*2	IOP*4
SCIF with FIFO	RxD2/SCPT[4]	Z	ZI*6	Z	IZ*5
	TxD2/SCPT[4]	Z	ZO*6	ZK*2	OZ*5
	SCK2/SCPT[5]	V	ZP*2	ZK*2	IOP*4
	$\overline{\text{RTS2}}$/SCPT[6]	V	OP*2	ZK*2	OP*2
	$\overline{\text{CTS2}}$/IRQ5/SCPT[7]	V*7	ZI*6	I	I
Port	$\overline{\text{AUDSYNC}}$/PTE[7]	OV	OP*2	OK*2	OP*2
	$\overline{\text{CE2B}}$/PTE[5]	V	OP*2	ZH*10 K*2	OP*2
	$\overline{\text{CE2A}}$/PTE[4]	V	OP*2	ZH*10 K*2	OP*2
	TDO/PTE[0]	OV	OP*2	OK*2	OP*2
	$\overline{\text{IOIS16}}$/PTG[7]	V	I	Z	I
	PTG[5:0]	V	I	Z	I
	AUDCK/PHT[6]	V	I	Z	I
	$\overline{\text{ADTRG}}$/PTH[5]	V*7	I	IZ	I
	$\overline{\text{WAKEUP}}$/PTD[3]	V	OP*2	OK*2	OP*2
	$\overline{\text{RESETOUT}}$/PTD[2]	O	OP*2	ZK*2	OP*2
	AUDATA[3:0]/PTG[3:0]	OV	OK	OK	OK
	CKIO2/PTG[4]	V	OI	OZ	OI
	$\overline{\text{ASEBRKAK}}$/PTG[5]	OV	OI	OZ	OI
	$\overline{\text{ASEMD0}}$/PTG[6]	I	I	Z	I
	PTJ[1]	H	OP*2	ZOK*3	OP*2
	PTE[1]	V	OP*2	ZOK*3	OP*2
	PTE[6]	V	OP*2	ZOK*3	OP*2
	PTE[3]	V	OP*2	ZOK*3	OP*2
	PTJ[4]	H	OP*2	ZOK*3	OP*2
	PTJ[5]	H	OP*2	ZOK*3	OP*2

L: Low level output
Z: High impedance
P: Input or output depending on register setting
K: Input pin is high impedance, output pin holds its state
V: I/O buffer off, pull-up MOS on

Notes:
1. Depending on the clock mode (MD2–MD0 setting).
2. K or P when the port function is used.
3. K or P when the port function is used. Z or O when the port function is not us depending on register setting.
4. K or P when the port function is used. I or O when the port function is not use depending on register setting.
5. Depending on register setting.
6. I or O when the port function is used.
7. Input Schmitt buffers of IRQ[5.0] and ADTRG on; other input buffers off.
8. O when DA output is enabled; otherwise I depending on register setting.
9. In standby mode, Z or L depending on register setting.
10. In standby mode, Z or H depending on register setting.
11. O when DA output is enabled; Z otherwise.

MD4, MD3	196, 195	D6, A7	I	Operating mode pin (area width)
MD2 to MD0	2, 1, 144	C2, D2,G19	I	Operating mode pin (cloc
$\overline{\text{RAS3L}}$/PTJ[0]	106	U18	I/O	RAS (SDRAM) / I/O port
PTJ[1]	107	U19	I/O	I/O port
$\overline{\text{CE2A}}$/PTE[4]	103	V17	I/O	PCMCIA CE2A / I/O port
$\overline{\text{CE2B}}$/PTE[5]	104	V16	I/O	PCMCIA CE2B / I/O port
RXD0/SCPT[0]	171	B13	I	Serial port 0 data input / i
RXD1/SCPT[2]	172	C13	I	Serial port 1 data input / i
RXD2/SCPT[4]	174	B12	I	Serial port 2 data input / i
TXD0/SCPT[0]	164	C15	O	Serial port 0 data output /
TXD1/SCPT[2]	166	A14	O	Serial port 1 data output /
TXD2/SCPT[4]	168	C14	O	Serial port 2 data output /
SCK0/SCPT[1]	165	D15	I/O	Serial port 0 clock input/o port
SCK1/SCPT[3]	167	B14	I/O	Serial port 1 clock input/o port
SCK2/SCPT[5]	169	D14	I/O	Serial port 2 clock input/o port
$\overline{\text{RTS2}}$/SCPT[6]	170	A13	I/O	Serial port 2 transfer requ port
STATUS1/PTJ[7]	158	B17	I/O	Processor state / I/O port
STATUS0/PTJ[6]	157	B16	I/O	Processor state / I/O port

		T5, U5, W5, W4, V5, V3, V4		
D31 to D24/ PTB[7] to PTB[0]	13 to 18, 20, 22	F4, G1, G2, G3, G4, H1, H3, J1	I/O	Data bus / I/O port
D23 to D16/ PTA[7] to PTA[0]	23 to 26, 28, 30 to 32	J2, J4, J3, K2, K1, L2, L1, M4	I/O	Data bus / I/O port
D15 to D0	34, 36 to 44, 46, 48 to 52	M2, N4, N3, N2, N1, P4, P3, P2, P1, R4, T4, T3, T1, R2, U2,T2	I/O	Data bus
$\overline{\text{MCS}}$[7:0]/ PTC[7:0]/ PINT[7:0]	177 to 180,185 to 188	B11, D11, C11, B10, D9, B9, A9, D8	I/O	Mask ROM chip select / I/O port interrupt request
$\overline{\text{WAKEUP}}$/PTD[3]	182	D10	I/O	Wakeup / I/O port
$\overline{\text{RESETOUT}}$/ PTD[2]	184	C9	I/O	Reset output / I/O port
DRAK0/PTD[1]	189	C8	I/O	DMA control pin / I/O port
DRAK1/PTD[0]	190	B8	I/O	DMA control pin / I/O port
$\overline{\text{DREQ0}}$/PTD[4]	191	A8	I	DMA transfer request 0 / in
$\overline{\text{DREQ1}}$/PTD[6]	192	D7	I	DMA transfer request 1 / in
AN[5:0]/PTL[5:0]	204 to 199	C4, A5, D4, C5, D5, A6	I	Analog input pin / input por
AN[7:6]/DA[1:0]/ PTL[7:6]	207, 206	B3, B5	I/O	Analog I/O pin / input port
$\overline{\text{CS6}}$/$\overline{\text{CE1B}}$	102	V15	O	Chip select 6 / PCMCIA CE
$\overline{\text{CS5}}$/$\overline{\text{CE1A}}$/ PTK[3]	101	W16	I/O	Chip select 5 / PCMCIA CE port
$\overline{\text{CS4}}$/PTK[2]	100	U16	I/O	Chip select 4 / I/O port
$\overline{\text{CS3}}$/PTK[1]	99	W15	I/O	Chip select 3 / I/O port

CASU/PTJ[3]	110	T17	I/O	CAS(SDRAM) / I/O port
CASL/PTJ[2]	108	R18	I/O	CAS(SDRAM) / I/O port
DACK0/PTD[5]	114	R16	I/O	DMA transfer strobe 0 / I/
DACK1/PTD[7]	115	P19	I/O	DMA transfer strobe 1 / I/
RD	88	T13	O	Read strobe pin
WE0/ DQMLL	89	U13	O	D7–D0 select signal/ DQI
WE1/DQMLU/WE	90	V13	O	D15–D8 select signal / DQM(SDRAM)/ PCMCIA WE signal
WE2/DQMUL/ ICIORD/PTK[6]	91	W13	I/O	D23–D16 select signal / DQM(SDRAM) / PCMCIA signal / I/O port
WE3/DQMUU/ ICIOWR/PTK[7]	92	T14	I/O	D31–D24 select signal /DQM(SDRAM) / PCMCI signal / I/O port
RD/WR	93	U14	O	Read/write select signal
AUDSYNC/ PTE[7]	94	V14	I/O	AUD synchronous I/O po
PTE[6]	116	P18	I/O	I/O port
PTE[3]	117	P17	I/O	I/O port
RAS3U/PTE[2]	118	P16	I/O	RAS(SDRAM) / I/O port
PTE[1]	119	N19	I/O	I/O port
TDO/PTE[0]	120	N18	I/O	Test data output I/O port
RESETM	124	M18	I	Manual reset input
ADTRG/PTH[5]	125	M17	I	ADC trigger request / Inp
IOIS16/PTG[7]	126	M16	I	I/O for PC card / input po
ASMD0/PTG[6]	127	L19	I	ASE mode / input port
ASEBRKAK/ PTG[5]	128	L18	I	ASE break accept / input

PTG[1]				
AUDATA[0]/ PTG[0]	135	J18	I	AUD data / input port
$\overline{\text{TRST}}$/PTF[7]/ PINT[15]	136	J19	I	Test reset / input port / por request
TMS/PTF[6]/ PINT[14]	137	H16	I	Test mode switch / input p interrupt request
TDI/PTF[5]/ PINT[13]	138	H17	I	Test data input / input port interrupt request
TCK/PTF[4]/ PINT[12]	139	H18	I	Test clock / input port / por request
$\overline{\text{IRLS}}$[3:0]/ PTF[3:0]/ PINT[11:8]	140 to 143	H19, G16, G17, G18	I	External interrupt request / / port interrupt request
AUDCK/PTH[6]	151	D16	I	AUD clock / input port
$\overline{\text{WAIT}}$	123	M19	I	Hardware wait request
$\overline{\text{BREQ}}$	122	N16	I	Bus request
$\overline{\text{BACK}}$	121	N17	O	Bus acknowledge
$\overline{\text{IRQOUT}}$	160	A16	O	Interrupt / refresh request
$\overline{\text{RESETP}}$	193	C7	I	Power-on reset input
NMI	7	C3	I	Nonmaskable interrupt req
IRQ[3:0]/$\overline{\text{IRL}}$[3:0]/ PTH[3:0]	11 to 8	F2, F1, E4, E3	I	External interrupt request / interrupt source / input por
IRQ4/PTH[4]	12	F3	I	External interrupt request /
$\overline{\text{CTS2}}$/IRQ5/ SCPT[7]	176	A11	I	Serial port 2 transfer enabl external interrupt request /
TCLK/PTH[7]	159	B15	I/O	Clock I/O (for TMU/RTC) /
EXTAL	156	D18	I	External clock / crystal osc
XTAL	155	C18	O	Crystal oscillator pin
CAP1	146	F17	—	External capacitance pin (f
CAP2	149	E16	—	External capacitance pin (f

CA	194	B7	I	Setting hardware standby
$V_{CC}Q$	21, 35, 47, 59, 71, 85, 97, 111, 163, 183	H4, M1, R1, U3, V8, U15, R19, C17, A10, U12	Power supply	Power supply (3.3 V)
V_{CC}–RTC	3	E2	Power supply	RTC oscillator power sup (2.0/1.9/1.8/1.7 V)
V_{CC}–PLL1 V_{CC}–PLL2	145 150	F16, E17	Power supply	PLL power supply (2.0/1.
AV_{CC}	205	A4	Power supply	Analog power supply (3.3
$V_{SS}Q$	19, 33, 45, 57, 69, 83, 95, 109, 161, 181	H2, M3, R3, T7, U4, W11, W14, T19, C16, C10	Power supply	Power supply (0 V)
V_{CC}	29, 81, 134, 154, 175	L3, L4, U11, T11, J17, J16, E18, C19, C12, D12	Power supply	Internal power supply (2.0/1.9/1.8/1.7 V)
V_{SS}	27, 79, 132, 152, 153, 173	K3, K4, U10, T10, K17, K16, E19, D17, D19, A12, D13	Power supply	Internal power supply (0 V
V_{SS}–RTC	6	E1	Power supply	RTC-oscillator power sup
V_{SS}–PLL1 V_{SS}–PLL2	147 148	F18 F19	Power supply	PLL power supply (0 V)
AV_{SS}	198, 208	B6, B4	Power supply	Analog power supply (0 V

Note: Except in hardware standby mode, power must be supplied constantly to all po pins. In hardware standby mode, power must be supplied to Vcc-RTC and Vss- least.

— CAP2: Leave unconnected
— V_{CC}–PLL2: Power supply (2.0/1.9/1.8/1.7 V)
— V_{SS}–PLL2: Power supply (0 V)

- When on-chip crystal oscillator is not used
  — XTAL: Leave unconnected
- When EXTAL pin is not used
  — EXTAL: Pull up (3.3 V)
- When A/D converter is not used
  — AN[7:0]: Leave unconnected
  — AV_{CC}: Power supply (3.3 V)
  — AV_{SS}: Power supply (0 V)
- When UDI is not used
  — ASEMD0: Pull up (3.3 V)

RD/$\overline{WR}$	R	High	High	High	High
	W	Low	Low	Low	Low
$\overline{BS}$		Enabled	Enabled	Enabled	Enal
$\overline{RAS3U}$/PTE[2]		High	High	High	High
$\overline{RAS3L}$/PTJ[0]		High	High	High	High
$\overline{CASL}$/PTJ[2]		High	High	High	High
$\overline{CASU}$/PTJ[3]		High	High	High	High
$\overline{WE0}$/DQMLL	R	High	High	High	High
	W	Low	Low	High	Low
$\overline{WE1}$/DQMLU/$\overline{WE}$	R	High	High	High	High
	W	High	High	Low	Low
$\overline{WE2}$/DQMUL/ $\overline{ICIORD}$/PTK[6]	R	High	High	High	High
	W	High	High	High	High
$\overline{WE3}$/DQMUU/ $\overline{ICIOWR}$/PTK[7]	R	High	High	High	High
	W	High	High	High	High
$\overline{CE2A}$/PTE[4]		High	High	High	High
$\overline{CE2B}$/PTE[5]		High	High	High	High
CKE/PTK[5]		Disabled	Disabled	Disabled	Disa
$\overline{WAIT}$		Enabled*1	Enabled*1	Enabled*1	Enal
$\overline{IOIS16}$/PTG[7]		Disabled	Disabled	Disabled	Disa
A25 to A0		Address	Address	Address	Addr
D7 to D0		Valid data	Valid data	Invalid data	Valid
D15 to D8		High-Z*2	Invalid data	Valid data	Valid
D31 to D16		High-Z*2	High-Z*2	High-Z*2	High

$\overline{BS}$		Enabled	Enabled	Enabled	Enabled	Enabled	Enabled
$\overline{RAS3U}$/PTE[2]		High	High	High	High	High	High
$\overline{RAS3L}$/PTJ[0]		High	High	High	High	High	High
$\overline{CASL}$/PTJ[2]		High	High	High	High	High	High
$\overline{CASU}$/PTJ[3]		High	High	High	High	High	High
$\overline{WE0}$/DQMLL	R	High	High	High	High	High	High
	W	Low	High	High	High	Low	High
$\overline{WE1}$/DQMLU/$\overline{WE}$	R	High	High	High	High	High	High
	W	High	Low	High	High	Low	High
$\overline{WE2}$/DQMUL/ $\overline{ICIORD}$/PTK[6]	R	High	High	High	High	High	High
	W	High	High	Low	High	High	Low
$\overline{WE3}$/DQMUU/ $\overline{ICIOWR}$/PTK[7]	R	High	High	High	High	High	High
	W	High	High	High	Low	High	Low
$\overline{CE2A}$/PTE[4]		High	High	High	High	High	High
$\overline{CE2B}$/PTE[5]		High	High	High	High	High	High
CKE/PTK[5]		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
$\overline{WAIT}$		Enabled*1	Enabled*1	Enabled*1	Enabled*1	Enabled*1	Enabled*1
$\overline{IOIS16}$/PTG[7]		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
A25 to A0		Address	Address	Address	Address	Address	Address
D7 to D0		Valid data	Invalid data	Invalid data	Invalid data	Valid data	Invalid data
D15 to D8		Invalid data	Valid data	Invalid data	Invalid data	Valid data	Invalid data
D23 to D16		Invalid data	Invalid data	Valid data	Invalid data	Invalid data	Valid data
D31 to D24		Invalid data	Invalid data	Invalid data	Valid data	Invalid data	Valid data

Notes: 1. Disabled when WCR2 register wait setting is 0.

2. Unused data pins should be switched to the port function, or pulled up.

	W	Low	Low	Low	Low
$\overline{BS}$		Enabled	Enabled	Enabled	Enal
$\overline{RAS3U}$/PTE[2]		High	High	High	High
$\overline{RAS3L}$/PTJ[0]		High	High	High	High
$\overline{CASL}$/PTJ[2]		High	High	High	High
$\overline{CASU}$/PTJ[3]		High	High	High	High
$\overline{WE0}$/DQMLL	R	High	High	High	High
	W	Low	High	Low	Low
$\overline{WE1}$/DQMLU/$\overline{WE}$	R	High	High	High	High
	W	High	Low	High	Low
$\overline{WE2}$/DQMUL/ $\overline{ICIORD}$/PTK[6]	R	High	High	High	High
	W	High	High	High	High
$\overline{WE3}$/DQMUU/ $\overline{ICIOWR}$/PTK[7]	R	High	High	High	High
	W	High	High	High	High
$\overline{CE2A}$/PTE[4]		High	High	High	High
$\overline{CE2B}$/PTE[5]		High	High	High	High
CKE/PTK[5]		Disabled	Disabled	Disabled	Disa
$\overline{WAIT}$		Enabled*1	Enabled*1	Enabled*1	Enal
$\overline{IOIS16}$/PTG[7]		Disabled	Disabled	Disabled	Disa
A25 to A0		Address	Address	Address	Addı
D7 to D0		Valid data	Invalid data	Valid data	Valic
D15 to D8		High-Z*2	Valid data	Invalid data	Valic
D31 to D16		High-Z*2	High-Z*2	High-Z*2	High

$\overline{BS}$		Enabled	Enabled	Enabled	Enabled	Enabled	Enabled
$\overline{RAS3U}$/PTE[2]		High	High	High	High	High	High
$\overline{RAS3L}$/PTJ[0]		High	High	High	High	High	High
$\overline{CASL}$/PTJ[2]		High	High	High	High	High	High
$\overline{CASU}$/PTJ[3]		High	High	High	High	High	High
$\overline{WE0}$/DQMLL	R	High	High	High	High	High	High
	W	High	High	High	Low	High	Low
$\overline{WE1}$/DQMLU/$\overline{WE}$	R	High	High	High	High	High	High
	W	High	High	Low	High	High	Low
$\overline{WE2}$/DQMUL/ $\overline{ICIORD}$/PTK[6]	R	High	High	High	High	High	High
	W	High	Low	High	High	Low	High
$\overline{WE3}$/DQMUU/ $\overline{ICIOWR}$/PTK[7]	R	High	High	High	High	High	High
	W	Low	High	High	High	Low	High
$\overline{CE2A}$/PTE[4]		High	High	High	High	High	High
$\overline{CE2B}$/PTE[5]		High	High	High	High	High	High
CKE/PTK[5]		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
$\overline{WAIT}$		Enabled*1	Enabled*1	Enabled*1	Enabled*1	Enabled*1	Enabled*1
$\overline{IOIS16}$/PTG[7]		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
A25 to A0		Address	Address	Address	Address	Address	Address
D7 to D0		Invalid data	Invalid data	Invalid data	Valid data	Invalid data	Valid data
D15 to D8		Invalid data	Invalid data	Valid data	Invalid data	Invalid data	Valid data
D23 to D16		Invalid data	Valid data	Invalid data	Invalid data	Valid data	Invalid data
D31 to D24		Valid data	Invalid data	Invalid data	Invalid data	Valid data	Invalid data

Notes: 1. Disabled when WCR2 register wait setting is 0.

2. Unused data pins should be switched to the port function, or pulled up.

	W	—	—	—	—
$\overline{BS}$		Enabled	Enabled	Enabled	Enal
$\overline{RAS3U}$/PTE[2]		High	High	High	High
$\overline{RAS3L}$/PTJ[0]		High	High	High	High
$\overline{CASL}$/PTJ[2]		High	High	High	High
$\overline{CASU}$/PTJ[3]		High	High	High	High
$\overline{WE0}$/DQMLL	R	High	High	High	High
	W	—	—	—	—
$\overline{WE1}$/DQMLU/$\overline{WE}$	R	High	High	High	High
	W	—	—	—	—
$\overline{WE2}$/DQMUL/ $\overline{ICIORD}$/PTK[6]	R	High	High	High	High
	W	—	—	—	—
$\overline{WE3}$/DQMUU/ $\overline{ICIOWR}$/PTK[7]	R	High	High	High	High
	W	—	—	—	—
$\overline{CE2A}$/PTE[4]		High	High	High	High
$\overline{CE2B}$/PTE[5]		High	High	High	High
CKE/PTK[5]		Disabled	Disabled	Disabled	Disa
$\overline{WAIT}$		Enabled*1	Enabled*1	Enabled*1	Enal
$\overline{IOIS16}$/PTG[7]		Disabled	Disabled	Disabled	Disa
A25 to A0		Address	Address	Address	Addı
D7 to D0		Valid data	Valid data	Invalid data	Valic
D15 to D8		High-Z*2	Invalid data	Valid data	Valic
D31 to D16		High-Z*2	High-Z*2	High-Z*2	High

$\overline{BS}$		Enabled	Enabled	Enabled	Enabled	Enabled	Enabled
$\overline{RAS3U}$/PTE[2]		High	High	High	High	High	High
$\overline{RAS3L}$/PTJ[0]		High	High	High	High	High	High
$\overline{CASL}$/PTJ[2]		High	High	High	High	High	High
$\overline{CASU}$/PTJ[3]		High	High	High	High	High	High
$\overline{WE0}$/DQMLL	R	High	High	High	High	High	High
	W	—	—	—	—	—	—
$\overline{WE1}$/DQMLU/$\overline{WE}$	R	High	High	High	High	High	High
	W	—	—	—	—	—	—
$\overline{WE2}$/DQMUL/ $\overline{ICIORD}$/PTK[6]	R	High	High	High	High	High	High
	W	—	—	—	—	—	—
$\overline{WE3}$/DQMUU/ $\overline{ICIOWR}$/PTK[7]	R	High	High	High	High	High	High
	W	—	—	—	—	—	—
$\overline{CE2A}$/PTE[4]		High	High	High	High	High	High
$\overline{CE2B}$/PTE[5]		High	High	High	High	High	High
CKE/PTK[5]		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
$\overline{WAIT}$		Enabled*1	Enabled*1	Enabled*1	Enabled*1	Enabled*1	Enabled*1
$\overline{IOIS16}$/PTG[7]		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
A25 to A0		Address	Address	Address	Address	Address	Address
D7 to D0		Valid data	Invalid data	Invalid data	Invalid data	Valid data	Invalid data
D15 to D8		Invalid data	Valid data	Invalid data	Invalid data	Valid data	Invalid data
D23 to D16		Invalid data	Invalid data	Valid data	Invalid data	Invalid data	Valid data
D31 to D24		Invalid data	Invalid data	Invalid data	Valid data	Invalid data	Valid data

Notes: 1. Disabled when WCR2 register wait setting is 0.

2. Unused data pins should be switched to the port function, or pulled up.

	W	—	—	—	—
$\overline{BS}$		Enabled	Enabled	Enabled	Enal
$\overline{RAS3U}$/PTE[2]		High	High	High	High
$\overline{RAS3L}$/PTJ[0]		High	High	High	High
$\overline{CASL}$/PTJ[2]		High	High	High	High
$\overline{CASU}$/PTJ[3]		High	High	High	High
$\overline{WE0}$/DQMLL	R	High	High	High	High
	W	—	—	—	—
$\overline{WE1}$/DQMLU/$\overline{WE}$	R	High	High	High	High
	W	—	—	—	—
$\overline{WE2}$/DQMUL/ $\overline{ICIORD}$/PTK[6]	R	High	High	High	High
	W	—	—	—	—
$\overline{WE3}$/DQMUU/ $\overline{ICIOWR}$/PTK[7]	R	High	High	High	High
	W	—	—	—	—
$\overline{CE2A}$/PTE[4]		High	High	High	High
$\overline{CE2B}$/PTE[5]		High	High	High	High
CKE/PTK[5]		Disabled	Disabled	Disabled	Disa
$\overline{WAIT}$		Enabled*1	Enabled*1	Enabled*1	Enal
$\overline{IOIS16}$/PTG[7]		Disabled	Disabled	Disabled	Disa
A25 to A0		Address	Address	Address	Addı
D7 to D0		Valid data	Invalid data	Valid data	Valic
D15 to D8		High-Z*2	Valid data	Invalid data	Valic
D31 to D16		High-Z*2	High-Z*2	High-Z*2	High

$\overline{BS}$		Enabled	Enabled	Enabled	Enabled	Enabled	Enabled
$\overline{RAS3U}$/PTE[2]		High	High	High	High	High	High
$\overline{RAS3L}$/PTJ[0]		High	High	High	High	High	High
$\overline{CASL}$/PTJ[2]		High	High	High	High	High	High
$\overline{CASU}$/PTJ[3]		High	High	High	High	High	High
$\overline{WE0}$/DQMLL	R	High	High	High	High	High	High
	W	—	—	—	—	—	—
$\overline{WE1}$/DQMLU/$\overline{WE}$	R	High	High	High	High	High	High
	W	—	—	—	—	—	—
$\overline{WE2}$/DQMUL/ $\overline{ICIORD}$/PTK[6]	R	High	High	High	High	High	High
	W	—	—	—	—	—	—
$\overline{WE3}$/DQMUU/ $\overline{ICIOWR}$/PTK[7]	R	High	High	High	High	High	High
	W	—	—	—	—	—	—
$\overline{CE2A}$/PTE[4]		High	High	High	High	High	High
$\overline{CE2B}$/PTE[5]		High	High	High	High	High	High
CKE/PTK[5]		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
$\overline{WAIT}$		Enabled*1	Enabled*1	Enabled*1	Enabled*1	Enabled*1	Enabled*1
$\overline{IOIS16}$/PTG[7]		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
A25 to A0		Address	Address	Address	Address	Address	Address
D7 to D0		Invalid data	Invalid data	Invalid data	Valid data	Invalid data	Valid data
D15 to D8		Invalid data	Invalid data	Valid data	Invalid data	Invalid data	Valid data
D23 to D16		Invalid data	Valid data	Invalid data	Invalid data	Valid data	Invalid data
D31 to D24		Valid data	Invalid data	Invalid data	Invalid data	Valid data	Invalid data

Notes: 1. Disabled when WCR2 register wait setting is 0.

2. Unused data pins should be switched to the port function, or pulled up.

RD/$\overline{WR}$	R	High	High	High	High	High	High
	W	Low	Low	Low	Low	Low	Low
$\overline{BS}$		Enabled	Enabled	Enabled	Enabled	Enabled	Enabled
$\overline{RAS3U}$/PTE[2]		High/Low*1	High/Low*1	High/Low*1	High/Low*1	High/Low*1	High/Lov
$\overline{RAS3L}$/PTJ[0]		Low/High*1	Low/High*1	Low/High*1	Low/High*1	Low/High*1	Low/High
$\overline{CASL}$/PTJ[2]		Low	Low	Low	Low	Low	Low
$\overline{CASU}$/PTJ[3]		High	High	High	High	High	High
$\overline{WE0}$/DQMLL	R	Low	High	High	High	Low	High
	W	Low	High	High	High	Low	High
$\overline{WE1}$/DQMLU/$\overline{WE}$	R	High	Low	High	High	Low	High
	W	High	Low	High	High	Low	High
$\overline{WE2}$/DQMUL/ $\overline{ICIORD}$/PTK[6]	R	High	High	Low	High	High	Low
	W	High	High	Low	High	High	Low
$\overline{WE3}$/DQMUU/ $\overline{ICIOWR}$/PTK[7]	R	High	High	High	Low	High	Low
	W	High	High	High	Low	High	Low
$\overline{CE2A}$/PTE[4]		High	High	High	High	High	High
$\overline{CE2B}$/PTE[5]		High	High	High	High	High	High
CKE/PTK[5]		High*2	High*2	High*2	High*2	High*2	High*2
$\overline{WAIT}$		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
$\overline{IOIS16}$/PTG[7]		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
A25 to A0		Address command	Address command	Address command	Address command	Address command	Address comman
D7 to D0		Valid data	Invalid data	Invalid data	Invalid data	Valid data	Invalid data
D15 to D8		Invalid data	Valid data	Invalid data	Invalid data	Valid data	Invalid data
D23 to D16		Invalid data	Invalid data	Valid data	Invalid data	Invalid data	Valid data
D31 to D24		Invalid data	Invalid data	Invalid data	Valid data	Invalid data	Valid data

Notes: 1. Lower 32-MB access/ Upper 32-MB access/64-MB access
2. Normally high. Low in self-refreshing.

RD/$\overline{WR}$	R	High	High	High	High	High	High
	W	Low	Low	Low	Low	Low	Low
$\overline{BS}$		Enabled	Enabled	Enabled	Enabled	Enabled	Enabled
$\overline{RAS3U}$/PTE[2]		High/Low*1	High/Low*1	High/Low*1	High/Low*1	High/Low*1	High/Low*
$\overline{RAS3L}$/PTJ[0]		Low/High*1	Low/High*1	Low/High*1	Low/High*1	Low/High*1	Low/High*
$\overline{CASL}$/PTJ[2]		Low	Low	Low	Low	Low	Low
$\overline{CASU}$/PTJ[3]		High	High	High	High	High	High
$\overline{WE0}$/DQMLL	R	High	High	High	Low	High	Low
	W	High	High	High	Low	High	Low
$\overline{WE1}$/DQMLU/$\overline{WE}$	R	High	High	Low	High	High	Low
	W	High	High	Low	High	High	Low
$\overline{WE2}$/DQMUL/ $\overline{ICIORD}$/PTK[6]	R	High	Low	High	High	Low	High
	W	High	Low	High	High	Low	High
$\overline{WE3}$/DQMUU/ $\overline{ICIOWR}$/PTK[7]	R	Low	High	High	High	Low	High
	W	Low	High	High	High	Low	High
$\overline{CE2A}$/PTE[4]		High	High	High	High	High	High
$\overline{CE2B}$/PTE[5]		High	High	High	High	High	High
CKE/PTK[5]		High*2	High*2	High*2	High*2	High*2	High*2
$\overline{WAIT}$		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
$\overline{IOIS16}$/PTG[7]		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
A25 to A0		Address command	Address command	Address command	Address command	Address command	Address command
D7 to D0		Invalid data	Invalid data	Invalid data	Valid data	Invalid data	Valid data
D15 to D8		Invalid data	Invalid data	Valid data	Invalid data	Invalid data	Valid data
D23 to D16		Invalid data	Valid data	Invalid data	Invalid data	Valid data	Invalid data
D31 to D24		Valid data	Invalid data	Invalid data	Invalid data	Valid data	Invalid data

Notes: 1. Lower 32-MB access/ Upper 32-MB access/64-MB access
2. Normally high. Low in self-refreshing.

$\overline{CS0}$ to $\overline{CS2}$, $\overline{CS5}$		Enabled	Enabled	High	Enabled	Enabled	Enabled	High
$\overline{RD}$	R	Low	Low	Low	Low	High	High	High
	W	High	High	High	High	High	High	High
RD/$\overline{WR}$	R	High	High	High	High	High	High	High
	W	Low	Low	Low	Low	Low	Low	Low
$\overline{BS}$		Enabled	Enabled	Enabled	Enabled	Enabled	Enabled	Enabl
$\overline{RAS3U}$/PTE[2]		High	High	High	High	High	High	High
$\overline{RAS3L}$/PTJ[0]		High	High	High	High	High	High	High
$\overline{CASL}$/PTJ[2]		High	High	High	High	High	High	High
$\overline{CASU}$/PTJ[3]		High	High	High	High	High	High	High
$\overline{WE0}$/DQMLL	R	High	High	High	High	High	High	High
	W	High	High	High	High	High	High	High
$\overline{WE1}$/DQMLU/$\overline{WE}$	R	High	High	High	High	High	High	High
	W	Low	Low	Low	Low	High	High	High
$\overline{WE2}$/DQMUL/ $\overline{ICIORD}$/PTK[6]	R	High	High	High	High	Low	Low	Low
	W	High	High	High	High	High	High	High
$\overline{WE3}$/DQMUU/ $\overline{ICIOWR}$/PTK[7]	R	High	High	High	High	High	High	High
	W	High	High	High	High	Low	Low	Low
$\overline{CE2A}$/PTE[4]		High	High	Low	Low	High	High	Low
$\overline{CE2B}$/PTE[5]		High	High	High	High	High	High	High
CKE/PTK[5]		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabl
$\overline{WAIT}$		Enabled*1	Enabled*1	Enabled*1	Enabled*1	Enabled*1	Enabled*1	Enabl
$\overline{IOIS16}$/PTG[7]		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Enabl
A25 to A0		Address	Address	Address	Address	Address	Address	Addre
D7 to D0		Valid data	Valid data	Invalid data	Valid data	Valid data	Valid data	Invalid data
D15 to D8		High-Z*2	Invalid data	Valid data	Valid data	High-Z*2	Invalid data	Valid data
D31 to D16		High-Z*2	High-Z*2	High-Z*2	High-Z*2	High-Z*2	High-Z*2	High-Z

	W	High	High	High	High	High	High	High
RD/$\overline{WR}$	R	High	High	High	High	High	High	High
	W	Low	Low	Low	Low	Low	Low	Low
$\overline{BS}$		Enabled	Enabled	Enabled	Enabled	Enabled	Enabled	Enable
$\overline{RAS3U}$/PTE[2]		High	High	High	High	High	High	High
$\overline{RAS3L}$/PTJ[0]		High	High	High	High	High	High	High
$\overline{CASL}$/PTJ[2]		High	High	High	High	High	High	High
$\overline{CASU}$/PTJ[3]		High	High	High	High	High	High	High
$\overline{WE0}$/DQMLL	R	High	High	High	High	High	High	High
	W	High	High	High	High	High	High	High
$\overline{WE1}$/DQMLU/$\overline{WE}$	R	High	High	High	High	High	High	High
	W	Low	Low	Low	Low	High	High	High
$\overline{WE2}$/DQMUL/ $\overline{ICIORD}$/PTK[6]	R	High	High	High	High	Low	Low	Low
	W	High	High	High	High	High	High	High
$\overline{WE3}$/DQMUU/ $\overline{ICIOWR}$/PTK[7]	R	High	High	High	High	High	High	High
	W	High	High	High	High	Low	Low	Low
$\overline{CE2A}$/PTE[4]		High	High	High	High	High	High	High
$\overline{CE2B}$/PTE[5]		High	High	Low	Low	High	High	Low
CKE/PTK[5]		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disable
$\overline{WAIT}$		Enabled*1	Enabled*1	Enabled*1	Enabled*1	Enabled*1	Enabled*1	Enable
$\overline{IOIS16}$/PTG[7]		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Enable
A25 to A0		Address	Address	Address	Address	Address	Address	Addres
D7 to D0		Valid data	Valid data	Invalid data	Valid data	Valid data	Valid data	Invalid data
D15 to D8		High-Z*2	Invalid data	Valid data	Valid data	High-Z*2	Invalid data	Valid data
D31 to D16		High-Z*2	High-Z*2	High-Z*2	High-Z*2	High-Z*2	High-Z*2	High-Z*

Notes: 1. Disabled when WCR2 register wait setting is 0.

2. Unused data pins should be switched to the port function, or pulled up.

$\overline{CS0}$ to $\overline{CS2}$, $\overline{CS5}$		Enabled	Enabled	High	Enabled	Enabled	Enabled	High
$\overline{RD}$	R	Low	Low	Low	Low	High	High	High
	W	High	High	High	High	High	High	High
RD/$\overline{WR}$	R	High	High	High	High	High	High	High
	W	Low	Low	Low	Low	Low	Low	Low
$\overline{BS}$		Enabled	Enabled	Enabled	Enabled	Enabled	Enabled	Enabl
$\overline{RAS3U}$/PTE[2]		High	High	High	High	High	High	High
$\overline{RAS3L}$/PTJ[0]		High	High	High	High	High	High	High
$\overline{CASL}$/PTJ[2]		High	High	High	High	High	High	High
$\overline{CASU}$/PTJ[3]		High	High	High	High	High	High	High
$\overline{WE0}$/DQMLL	R	High	High	High	High	High	High	High
	W	High	High	High	High	High	High	High
$\overline{WE1}$/DQMLU/$\overline{WE}$	R	High	High	High	High	High	High	High
	W	Low	Low	Low	Low	High	High	High
$\overline{WE2}$/DQMUL/ $\overline{ICIORD}$/PTK[6]	R	High	High	High	High	Low	Low	Low
	W	High	High	High	High	High	High	High
$\overline{WE3}$/DQMUU/ $\overline{ICIOWR}$/PTK[7]	R	High	High	High	High	High	High	High
	W	High	High	High	High	Low	Low	Low
$\overline{CE2A}$/PTE[4]		High	High	Low	Low	High	High	Low
$\overline{CE2B}$/PTE[5]		High	High	High	High	High	High	High
CKE/PTK[5]		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabl
$\overline{WAIT}$		Enabled*1	Enabled*1	Enabled*1	Enabled*1	Enabled*1	Enabled*1	Enabl
$\overline{IOIS16}$/PTG[7]		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabl
A25 to A0		Address	Address	Address	Address	Address	Address	Addre
D7 to D0		Valid data	Invalid data	Valid data	Valid data	Valid data	Invalid data	Valid data
D15 to D8		High-Z*2	Valid data	Invalid data	Valid data	High-Z*2	Valid data	Invalid data
D31 to D16		High-Z*2	High-Z*2	High-Z*2	High-Z*2	High-Z*2	High-Z*2	High-Z

RD	R	Low	Low	Low	Low	High	High	High
	W	High	High	High	High	High	High	High
RD/$\overline{WR}$	R	High	High	High	High	High	High	High
	W	Low	Low	Low	Low	Low	Low	Low
$\overline{BS}$		Enabled	Enabled	Enabled	Enabled	Enabled	Enabled	Enabled
$\overline{RAS3U}$/PTE[2]		High	High	High	High	High	High	High
$\overline{RAS3L}$/PTJ[0]		High	High	High	High	High	High	High
$\overline{CASL}$/PTJ[2]		High	High	High	High	High	High	High
$\overline{CASU}$/PTJ[3]		High	High	High	High	High	High	High
$\overline{WE0}$/DQMLL	R	High	High	High	High	High	High	High
	W	High	High	High	High	High	High	High
$\overline{WE1}$/DQMLU/$\overline{WE}$	R	High	High	High	High	High	High	High
	W	Low	Low	Low	Low	High	High	High
$\overline{WE2}$/DQMUL/ $\overline{ICIORD}$/PTK[6]	R	High	High	High	High	Low	Low	Low
	W	High	High	High	High	High	High	High
$\overline{WE3}$/DQMUU/ $\overline{ICIOWR}$/PTK[7]	R	High	High	High	High	High	High	High
	W	High	High	High	High	Low	Low	Low
$\overline{CE2A}$/PTE[4]		High	High	High	High	High	High	High
$\overline{CE2B}$/PTE[5]		High	High	Low	Low	High	High	Low
CKE/PTK[5]		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disable
$\overline{WAIT}$		Enabled*1	Enabled*1	Enabled*1	Enabled*1	Enabled*1	Enabled*1	Enable
$\overline{IOIS16}$/PTG[7]		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Enable
A25 to A0		Address	Address	Address	Address	Address	Address	Address
D7 to D0		Valid data	Invalid data	Valid data	Valid data	Valid data	Invalid data	Valid data
D15 to D8		High-Z*2	Valid data	Invalid data	Valid data	High-Z*2	Valid data	Invalid data
D31 to D16		High-Z*2	High-Z*2	High-Z*2	High-Z*2	High-Z*2	High-Z*2	High-Z*

Notes: 1. Disabled when WCR2 register wait setting is 0.

2. Unused data pins should be switched to the port function, or pulled up.

PTEL	CCN	L	FFFFFFF4	32	3
TTB	CCN	L	FFFFFFF8	32	3
TEA	CCN	L	FFFFFFFC	32	3
MMUCR	CCN	L	FFFFFFE0	32	3
BASRA	CCN	L	FFFFFFE4	32	3
BASRB	CCN	L	FFFFFFE8	32	3
CCR	CCN	L	FFFFFFEC	32	3
CCR2	CCN	I	40000B0	32	3
TRA	CCN	L	FFFFFFD0	32	3
EXPEVT	CCN	L	FFFFFFD4	32	3
INTEVT	CCN	L	FFFFFFD8	32	3
BARA	UBC	L	FFFFFFB0	32	3
BAMRA	UBC	L	FFFFFFB4	32	3
BBRA	UBC	L	FFFFFFB8	16	1
BARB	UBC	L	FFFFFFA0	32	3
BAMRB	UBC	L	FFFFFFA4	32	3
BBRB	UBC	L	FFFFFFA8	16	1
BDRB	UBC	L	FFFFFF90	32	3
BDMRB	UBC	L	FFFFFF94	32	3
BRCR	UBC	L	FFFFFF98	16	1
BETR	UBC	L	FFFFFF9C	16	1
BRSR	UBC	L	FFFFFFAC	32	3
BRDR	UBC	L	FFFFFFBC	32	3
FRQCR	CPG	I	FFFFFF80	16	1
STBCR	CPG	I	FFFFFF82	8	
STBCR2	CPG	I	FFFFFF88	8	
WTCNT	CPG	I	FFFFFF84	8	1

RTCSR	BSC	I	FFFFFF6E	16	16
RTCNT	BSC	I	FFFFFF70	16	16
RTCOR	BSC	I	FFFFFF72	16	16
RFCR	BSC	I	FFFFFF74	16	16
SDMR	BSC	I	FFFFD000–FFFFEFFF	—	8
MCSCR0	BSC	I	FFFFFF50	16	16
MCSCR1	BSC	I	FFFFFF52	16	16
MCSCR2	BSC	I	FFFFFF54	16	16
MCSCR3	BSC	I	FFFFFF56	16	16
MCSCR4	BSC	I	FFFFFF58	16	16
MCSCR5	BSC	I	FFFFFF5A	16	16
MCSCR6	BSC	I	FFFFFF5C	16	16
MCSCR7	BSC	I	FFFFFF5E	16	16
R64CNT	RTC	P	FFFFFEC0	8	8
RSECCNT	RTC	P	FFFFFEC2	8	8
RMINCNT	RTC	P	FFFFFEC4	8	8
RHRCNT	RTC	P	FFFFFEC6	8	8
RWKCNT	RTC	P	FFFFFEC8	8	8
RDAYCNT	RTC	P	FFFFFECA	8	8
RMONCNT	RTC	P	FFFFFECC	8	8
RYRCNT	RTC	P	FFFFFECE	8	8
RSECAR	RTC	P	FFFFFED0	8	8
RMINAR	RTC	P	FFFFFED2	8	8
RHRAR	RTC	P	FFFFFED4	8	8

IPRB	INTC	I	FFFFFEE4	16	1
TOCR	TMU	P	FFFFFE90	8	
TSTR	TMU	P	FFFFFE92	8	
TCOR0	TMU	P	FFFFFE94	32	3
TCNT0	TMU	P	FFFFFE98	32	3
TCR0	TMU	P	FFFFFE9C	16	1
TCOR1	TMU	P	FFFFFEA0	32	3
TCNT1	TMU	P	FFFFFEA4	32	3
TCR1	TMU	P	FFFFFEA8	16	1
TCOR2	TMU	P	FFFFFEAC	32	3
TCNT2	TMU	P	FFFFFEB0	32	3
TCR2	TMU	P	FFFFFEB4	16	1
TCPR2	TMU	P	FFFFFEB8	32	3
SCSMR	SCI	P	FFFFFE80	8	
SCBRR	SCI	P	FFFFFE82	8	
SCSCR	SCI	P	FFFFFE84	8	
SCTDR	SCI	P	FFFFFE86	8	
SCSSR	SCI	P	FFFFFE88	8	
SCRDR	SCI	P	FFFFFE8A	8	
SCSCMR	SCI	P	FFFFFE8C	8	
INTEVT2	INTC	I	4000000	32	3
IRR0	INTC	I	4000004	16	
IRR1	INTC	I	4000006	16	
IRR2	INTC	I	4000008	16	

DAR0	DMAC	P	4000024	32	16,32
DMATCR0	DMAC	P	4000028	32	16,32
CHCR0	DMAC	P	400002C	32	8,16,
SAR1	DMAC	P	4000030	32	16,32
DAR1	DMAC	P	4000034	32	16,32
DMATCR1	DMAC	P	4000038	32	16,32
CHCR1	DMAC	P	400003C	32	8,16,
SAR2	DMAC	P	4000040	32	16,32
DAR2	DMAC	P	4000044	32	16,32
DMATCR2	DMAC	P	4000048	32	16,32
CHCR2	DMAC	P	400004C	32	8,16,
SAR3	DMAC	P	4000050	32	16,32
DAR3	DMAC	P	4000054	32	16,32
DMATCR3	DMAC	P	4000058	32	16,32
CHCR3	DMAC	P	400005C	32	8,16,
DMAOR	DMAC	P	4000060	16	8,16
CMSTR	CMT	P	4000070	16	8,16
CMCSR	CMT	P	4000072	16	8,16,
CMCNT	CMT	P	4000074	16	8,16,
CMCOR	CMT	P	4000076	16	8,16,
ADDRAH	A/D	P	4000080	8	8,16
ADDRAL	A/D	P	4000082	8	8,16*
ADDRBH	A/D	P	4000084	8	8,16
ADDRBL	A/D	P	4000086	8	8,16*
ADDRCH	A/D	P	4000088	8	8,16

DACR	D/A	P	40000A4	8	8,1
PACR	PORT	P	4000100	16	16
PBCR	PORT	P	4000102	16	16
PCCR	PORT	P	4000104	16	16
PDCR	PORT	P	4000106	16	16
PECR	PORT	P	4000108	16	16
PFCR	PORT	P	400010A	16	16
PGCR	PORT	P	400010C	16	16
PHCR	PORT	P	400010E	16	16
PJCR	PORT	P	4000110	16	16
PKCR	PORT	P	4000112	16	16
PLCR	PORT	P	4000114	16	16
SCPCR	PORT	P	4000116	16	16
PADR	PORT	P	4000120	8	8
PBDR	PORT	P	4000122	8	8
PCDR	PORT	P	4000124	8	8
PDDR	PORT	P	4000126	8	8
PEDR	PORT	P	4000128	8	8
PFDR	PORT	P	400012A	8	8
PGDR	PORT	P	400012C	8	8
PHDR	PORT	P	400012E	8	8
PJDR	PORT	P	4000130	8	8
PKDR	PORT	P	4000132	8	8
PLDR	PORT	P	4000134	8	8
SCPDR	PORT	P	4000136	8	8

SCFDR1	IrDA	P	400014E	16	16
SCSMR2	SCIF	P	4000150	8	8
SCBRR2	SCIF	P	4000152	8	8
SCSCR2	SCIF	P	4000154	8	8
SCFTDR2	SCIF	P	4000156	8	8
SCSSR2	SCIF	P	4000158	16	16
SCFRDR2	SCIF	P	400015A	8	8
SCFCR2	SCIF	P	400015C	8	8
SCFDR2	SCIF	P	400015E	16	16
SDIR	UDI	I	4000200	16	16

Notes: 1. Modules:

CCN: Cache controller — UBC: User break controller
CPG: Clock pulse generator — BSC: Bus state controller
RTC: Realtime clock — INTC: Interrupt controller
TMU: Timer unit — SCI: Serial communication interface

2. Internal buses:

L: CPU, CCN, cache, TLB connected

I: BSC, cache, DMAC, INTC, CPG, and UDI connected

P: BSC and peripheral modules (RTC, TMU, SCI, SCIF, IrDA, A/D, D/A, DM PORT, CMT) connected

3. The access size shown is for control register access (read/write). An incorre will be obtained if a different size from that shown is used for access.
4. To exclude area 1 control registers from address translation by the MMU, se bits of the logical address to 101, to locate the registers in the P2 space.
5. With 16-bit access, it is not possible to read data in two registers simultaneo
6. With 32-bit access, it is possible to read data in the register at [accessed ad simultaneously.

SCSCR	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0
SCTDR								
SCSSR	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT
SCRDR								
SCSCMR	—	—	—	—	SDIR	SINV	—	SMIF
TOCR	—	—	—	—	—	—	—	TCOE
TSTR	—	—	—	—	—	STR2	STR1	STR0
TCOR0								
TCNT0								
TCR0	—	—	—	—	—	—	—	UNF
	—	—	UNIE	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0
TCOR1								
TCNT1								

TCR2	—	—	—	—	—	—	ICPF	UNF
	ICPE1	ICPE0	UNIE	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0
TCPR2								
R64CNT	—	1 Hz	2 Hz	4 Hz	8 Hz	16 Hz	32 Hz	64 Hz
RSECCNT	—	10 sec			1 sec			
RMINCNT	—	10 min			1 min			
RHRCNT	—	—	10 hours		1 hour			
RWKCNT	—	—	—	—	—	Day of week		
RDAYCNT	—	—	10 days		1 day			
RMONCNT	—	—	—	10 months	1 month			
RYRCNT	10 years				1 year			
RSECAR	ENB	10 sec			1 sec			
RMINAR	ENB	10 min			1 min			
RWKAR	ENB	—	—	—	—	Day of week		
RHRAR	ENB	—	10 hours		1 hour			
RDAYAR	ENB	—	10 days		1 day			
RMONAR	ENB	—	—	10 months	1 month			

	SCI				—	—	—	—
BCR1	PULA	PULD	HIZMEM	HIZCNT	ENDIAN	A0BST1	A0BST0	A5BST1
	A5BST0	A6BST1	A6BST0	DRAMTP2	DRAMTP1	DRAMTP0	A5PCM	A6PCM
BCR2	—	—	A6SZ1	A6SZ0	A5SZ1	A5SZ0	A4SZ1	A4SZ0
	A3SZ1	A3SZ0	A2SZ1	A2SZ0	—	—	—	—
WCR1	WAITSEL	—	A6IW1	A6IW0	A5IW1	A5IW0	A4IW1	A4IW0
	A3IW1	A3IW0	A2IW1	A2IW0	—	—	A0IW1	A0IW0
WCR2	A6W2	A6W1	A6W0	A5W2	A5W1	A5W0	A4W2	A4W1
	A4W0	A3W1	A3W0	A2W1	A2W0	A0W2	A0W1	A0W0
MCR	TPC1	TPC0	RCD1	RCD0	TRWL1	TRWL0	TRAS1	TRAS0
	RASD	AMX3	AMX2	AMX1	AMX0	RFSH	RMODE	—
PCR	A6W3	A5W3	—	—	A5TED2	A6TED2	A5TEH2	A6TEH2
	A5TED1	A5TED0	A6TED1	A6TED0	A5TEH1	A5TEH0	A6TEH1	A6TEH0
RTCSR	—	—	—	—	—	—	—	—
	CMF	CMIE	CKS2	CKS1	CKS0	OVF	OVIE	LMTS
RTCNT	—	—	—	—	—	—	—	—
RTCOR	—	—	—	—	—	—	—	—

WTCSR	TME	WT/IT	RSTS	WOVF	IOVF	CKS2	CKS1	CKS0
BDRB								
BDMRB								
BRCR	—	—	—	—	—	—	—	—
	—	—	BASMA	BASMB	—	—	—	—
	SCMFCA	SCMFCB	SCMFDA	SCMFDB	PCTE	PCBA	—	—
	DBEB	PCBB	—	—	SEQ	—	—	ETBE
BARB								
BAMRB	—	—	—	—	—	BASM	BAM	BAM
BBRB	—	—	—	—	—	—	—	—
	CDB1	CDB0	IDB1	IDB0	RWB1	RWB0	SZB1	SZB0
BARA								

BRSR	SVF	PID2	PID1	PID0	BSA27	BSA26	BSA25	BSA24
	BSA23	BSA22	BSA21	BSA20	BSA19	BSA18	BSA17	BSA16
	BSA15	BSA14	BSA13	BSA12	BSA11	BSA10	BSA9	BSA8
	BSA7	BSA6	BSA5	BSA4	BSA3	BSA2	BSA1	BSA0
BRDR	DVF	—	—	—	BDA27	BDA26	BDA25	BDA24
	BDA23	BDA22	BDA21	BDA20	BDA19	BDA18	BDA17	BDA16
	BDA15	BDA14	BDA13	BDA12	BDA11	BDA10	BDA9	BDA8
	BDA7	BDA6	BDA5	BDA4	BDA3	BDA2	BDA1	BDA0
TRA	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	—	—	—	—		
							—	—
EXPEVT	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	—	—				
INTEVT	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	—	—				
MMUCR	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	SV
	—	—	RC	RC	—	TF	IX	AT

							W3LOAD	W3LOCI
							W2LOAD	W2LOCI
PTEH								
							—	—
PTEL								
							—	V
	—	PR	PR	SZ	C	D	SH	—
TTB								
TEA								
INTEVT2								

	IRQ31S	IRQ30S	IRQ21S	IRQ20S	IRQ11S	IRQ10S	IRQ01S	IRQ00S
ICR2	PINT15S	PINT14S	PINT13S	PINT12S	PINT11S	PINT10S	PINT9S	PINT8S
	PINT7S	PINT6S	PINT5S	PINT4S	PINT3S	PINT2S	PINT1S	PINT0S
PINTER	PINT15E	PINT14E	PINT13E	PINT12E	PINT11E	PINT10E	PINT9E	PINT8E
	PINT7E	PINT6E	PINT5E	PINT4E	PINT3E	PINT2E	PINT1E	PINT0E
IPRC	IRQ3 level				IRQ2 level			
	IRQ1 level				IRQ0 level			
IPRD	PINT0 to 7 level				PINT8 to 15 level			
	IRQ5 level				IRQ4 level			
IPRE	DMAC level				IrDA level			
	SCIF level				A/D level			
SAR0								
DAR0								
DMATCR0	—	—	—	—	—	—	—	—

DAR1								
DMATCR1	—	—	—	—	—	—	—	—
CHCR1	—	—	—	—	—	—	—	—
	—	—	—	—	—	RL	AM	AL
	DM1	DM0	SM1	SM0	RS3	RS2	RS1	RS0
	—	DS	TM	TS1	TS0	IE	TE	DE
SAR2								
DAR2								
DMATCR2	—	—	—	—	—	—	—	—

DAR3								
DMATCR3	—	—	—	—	—	—	—	—
CHCR3	—	—	—	—	—	—	—	—
	—	—	—	DI	—	—	—	—
	DM1	DM0	SM1	SM0	RS3	RS2	RS1	RS0
	—	—	TM	TS1	TS0	IE	TE	DE
DMAOR	—	—	—	—	—	—	PR1	PR0
	—	—	—	—	—	AE	NMIF	DME
CMSTR	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	STR
CMCSR	—	—	—	—	—	—	—	—
	CMF	—	—	—	—	—	CKS1	CKS0
CMCNT								
CMCOR								
ADDRAH	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2
ADDRAL	AD1	AD0	—	—	—	—	—	—

ADCR	TRGE1	TRGE2	SCN	RESVD1	RESVD2	—	—	—	
DADR0									
DADR1									
DACR	DAOE1	DAOE0	DAE	—	—	—	—	—	
PACR	PA7MD1	PA7MD0	PA6MD1	PA6MD0	PA5MD1	PA5MD0	PA4MD1	PA4MD0	
	PA3MD1	PA3MD0	PA2MD1	PA2MD0	PA1MD1	PA1MD0	PA0MD1	PA0MD0	
PBCR	PB7MD1	PB7MD0	PB6MD1	PB6MD0	PB5MD1	PB5MD0	PB4MD1	PB4MD0	
	PB3MD1	PB3MD0	PB2MD1	PB2MD0	PB1MD1	PB1MD0	PB0MD1	PB0MD0	
PCDR	PC7MD1	PC7MD0	PC6MD1	PC6MD0	PC5MD1	PC5MD0	PC4MD1	PC4MD0	
	PC3MD1	PC3MD0	PC2MD1	PC2MD0	PC1MD1	PC1MD0	PC0MD1	PC0MD0	
PDCR	PD7MD1	PD7MD0	PD6MD1	PD6MD0	PD5MD1	PD5MD0	PD4MD1	PD4MD0	
	PD3MD1	PD3MD0	PD2MD1	PD2MD0	PD1MD1	PD1MD0	PD0MD1	PD0MD0	
PECR	PE7MD1	PE7MD0	PE6MD1	PE6MD0	PE5MD1	PE5MD0	PE4MD1	PE4MD0	
	PE3MD1	PE3MD0	PE2MD1	PE2MD0	PE1MD1	PE1MD0	PE0MD1	PE0MD0	
PFCR	PF7MD1	PF7MD0	PF6MD1	PF6MD0	PF5MD1	PF5MD0	PF4MD1	PF4MD0	
	PF3MD1	PF3MD0	PF2MD1	PF2MD0	PF1MD1	PF1MD0	PF0MD1	PF0MD0	

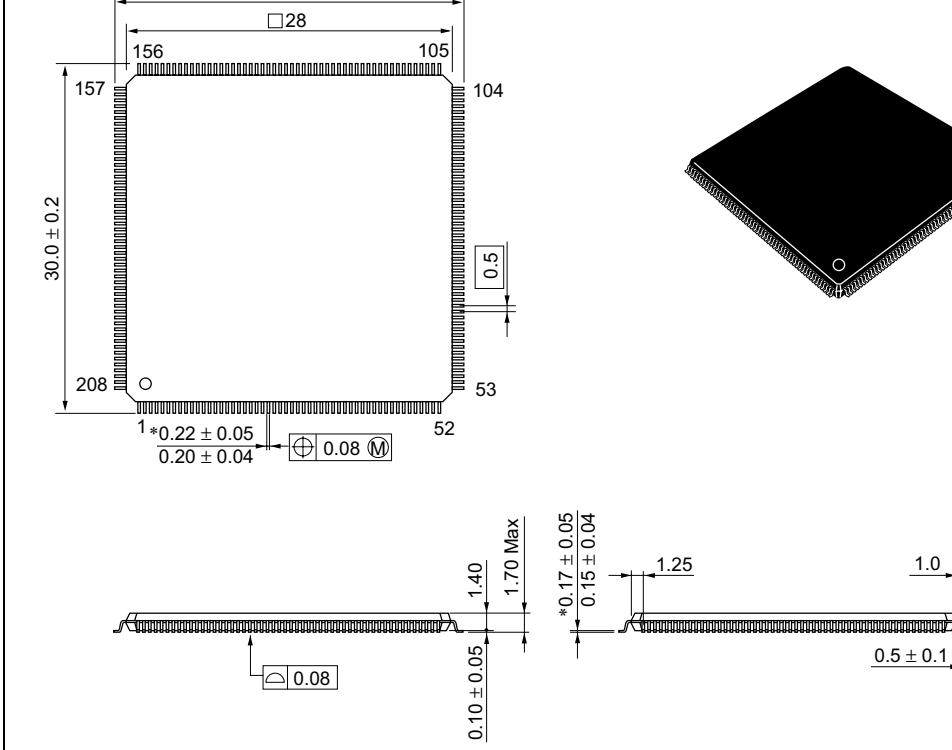
PJCR	PJ7MD1	PJ7MD0	PJ6MD1	PJ6MD0	PJ5MD1	PJ5MD0	PJ4MD1	PJ4MD0
	PJ3MD1	PJ3MD0	PJ2MD1	PJ2MD0	PJ1MD1	PJ1MD0	PJ0MD1	PJ0MD0
PKCR	PK7MD1	PK7MD0	PK6MD1	PK6MD0	PK5MD1	PK5MD0	PK4MD1	PK4MD0
	PK3MD1	PK3MD0	PK2MD1	PK2MD0	PK1MD1	PK1MD0	PK0MD1	PK0MD0
PLCR	PL7MD1	PL7MD0	PL6MD1	PL6MD0	PL5MD1	PL5MD0	PL4MD1	PL4MD0
	PL3MD1	PL3MD0	PL2MD1	PL2MD0	PL1MD1	PL1MD0	PL0MD1	PL0MD0
SCPCR	SCP7MD1	SCP7MD0	SCP6MD1	SCP6MD0	SCP5MD1	SCP5MD0	SCP4MD1	SCP4MD0
	SCP3MD1	SCP3MD0	SCP2MD1	SCP2MD0	SCP1MD1	SCP1MD0	SCP0MD1	SCP0MD0
PADR	PA7DT	PA6DT	PA5DT	PA4DT	PA3DT	PA2DT	PA1DT	PA0DT
PBDR	PB7DT	PB6DT	PB5DT	PB4DT	PB3DT	PB2DT	PB1DT	PB0DT
PCDR	PC7DT	PC6DT	PC5DT	PC4DT	PC3DT	PC2DT	PC1DT	PC0DT
PDDR	PD7DT	PD6DT	PD5DT	PD4DT	PD3DT	PD2DT	PD1DT	PD0DT
PEDR	PE7DT	PE6DT	PE5DT	PE4DT	PE3DT	PE2DT	PE1DT	PE0DT
PFDR	PF7DT	PF6DT	PF5DT	PF4DT	PF3DT	PF2DT	PF1DT	PF0DT
PGDR	PG7DT	PG6DT	PG5DT	PG4DT	PG3DT	PG2DT	PG1DT	PG0DT
PHDR	PH7DT	PH6DT	PH5DT	PH4DT	PH3DT	PH2DT	PH1DT	PH0DT
PJDR	PJ7DT	PJ6DT	PJ5DT	PJ4DT	PJ3DT	PJ2DT	PJ1DT	PJ0DT
PKDR	PK7DT	PK6DT	PK5DT	PK4DT	PK3DT	PK2DT	PK1DT	PK0DT
PLDR	PL7DT	PL6DT	PL5DT	PL4DT	PL3DT	PL2DT	PL1DT	PL0DT
SCPDR	SCP7DT	SCP6DT	SCP5DT	SCP4DT	SCP3DT	SCP2DT	SCP1DT	SCP0DT

	ER	TEND	TDFE	BRK	FER	PER	RDF	DR
SCFRDR1								
SCFCR1	RTRG1	RTRG0	TTRG1	TTRG0	MCE	TFRST	RFRST	LOOP
SCFDR1	—	—	—	T4	T3	T2	T1	T0
	—	—	—	R4	R3	R2	R1	R0
SCSMR2	—	CHR	PE	O/E	STOP	—	CKS1	CKS0
SCBRR2								
SCSCR2	TIE	RIE	TE	RE	—	—	CKE1	CKE0
SCFTDR2								
SCSSR2	PER3	PER2	PER1	PER0	FER3	FER2	FER1	FER0
	ER	TEND	TDFE	BRK	FER	PER	RDF	DR
SCFRDR2								
SCFCR2	RTRG1	RTRG0	TTRG1	TTRG0	MCE	TFRST	RFRST	LOOP
SCFDR2	—	—	—	T4	T3	T2	T1	T0
	—	—	—	R4	R3	R2	R1	R0

Legend

MMU: Memory management unit
UBC: User break controller
CPG: Clock pulse generator
BSC: Bus state controller
RTC: Realtime clock
INTC: Interrupt controller
TMU: Timer unit
SC1: Serial communication interface controller
IrDA: Serial communication interface with IrDA
SCIF: Serial communication interface with FIFO
CCN: Cache controller
DMAC: Direct memory access controller
ADC: Analog to Digital converter
DAC: Digital to Analog converter
PORT: Port controller
UDI: User debugging interface

		1.9±0.15 V	167 MHz	HD6417709SF167B	208-pin pl (FP-208C)
				HD6417709SBP167B	240-pin C (BP-240A)
		1.8+0.25 V 1.8–0.15 V	133 MHz	HD6417709SF133B	208-pin pl (FP-208C)
				HD6417709SBP133B	240-pin C (BP-240A)
		1.7+0.25 V 1.7–0.15 V	100 MHz	HD6417709SF100B	208-pin pl (FP-208C)
				HD6417709SBP100B	240-pin C (BP-240A)



Package Code	FP-208C
JEDEC	–
JEITA	Conforms
Mass (reference value)	2.7 g

*Dimension including the plating thickness
Base material dimension

Figure D.1 Package Dimensions (FP-208C)

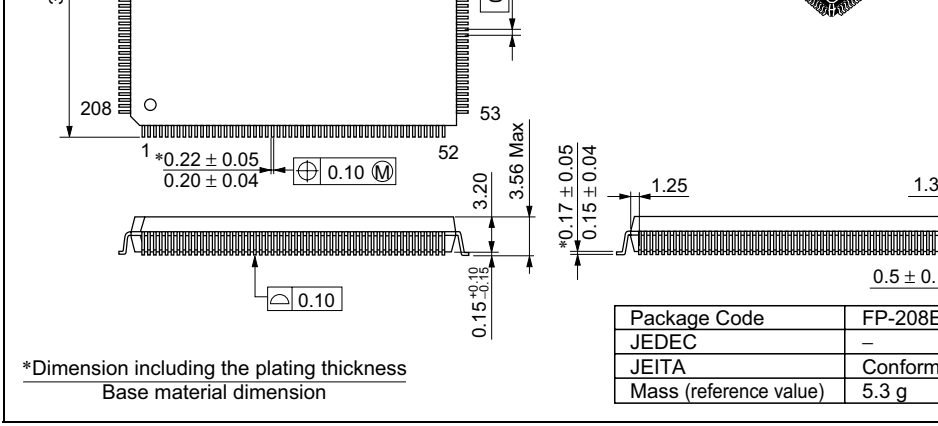
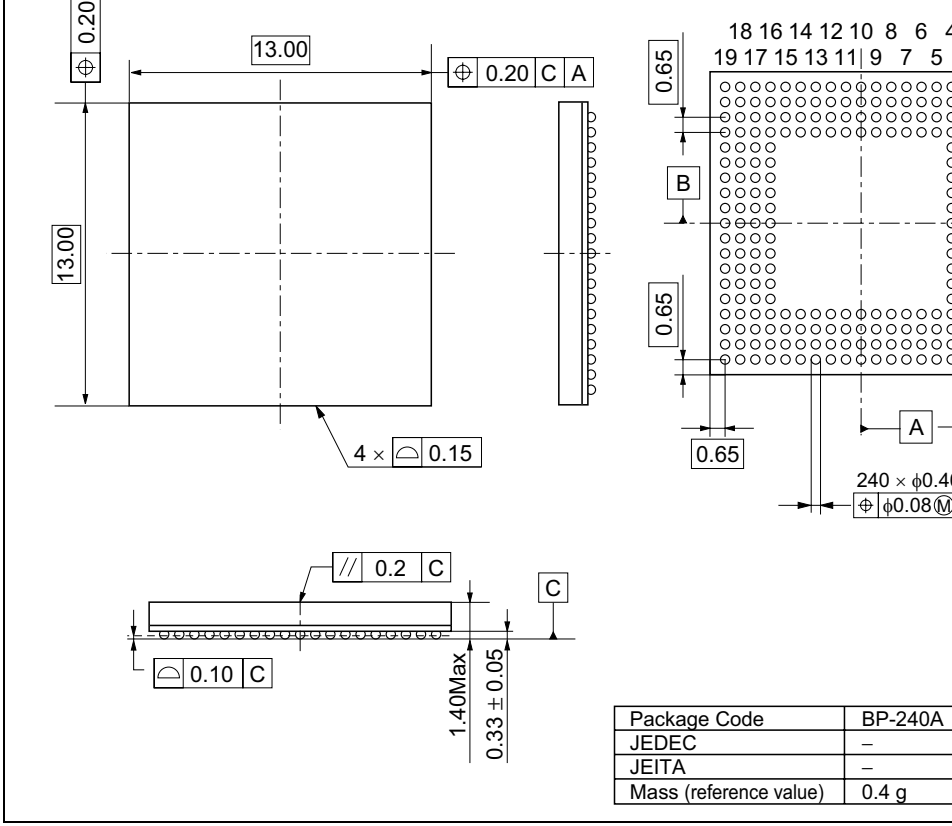


Figure D.2 Package Dimensions (FP-208E)



Package Code	BP-240A
JEDEC	–
JEITA	–
Mass (reference value)	0.4 g

Figure D.3 Package Dimensions (BP-240A)

SH7709S Group Hardware Manual

Publication Date:	1st Edition, September 2001
	Rev.5.00, September 18, 2003
Published by:	Sales Strategic Planning Div.
	Renesas Technology Corp.
Edited by:	Technical Documentation & Information Department
	Renesas Kodaira Semiconductor Co., Ltd.

Renesas Technology Corp. Sales Strategic Planning Div. Nippon Bldg., 2-6-2, Ohte-machi, Chiyoda-ku, Tokyo 100-000

RENESAS SALES OFFICES

http://www.

Renesas Technology America, Inc.
450 Holger Way, San Jose, CA 95134-1368, U.S.A
Tel: <1> (408) 382-7500 Fax: <1> (408) 382-7501

Renesas Technology Europe Limited.
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, United Kingdom
Tel: <44> (1628) 585 100, Fax: <44> (1628) 585 900

Renesas Technology Europe GmbH
Dornacher Str. 3, D-85622 Feldkirchen, Germany
Tel: <49> (89) 380 70 0, Fax: <49> (89) 929 30 11

Renesas Technology Hong Kong Ltd.
7/F., North Tower, World Finance Centre, Harbour City, Canton Road, Hong Kong
Tel: <852> 2265-6688, Fax: <852> 2375-6836

Renesas Technology Taiwan Co., Ltd.
FL 10, #99, Fu-Hsing N. Rd., Taipei, Taiwan
Tel: <886> (2) 2715-2888, Fax: <886> (2) 2713-2999

Renesas Technology (Shanghai) Co., Ltd.
26/F., Ruijin Building, No.205 Maoming Road (S), Shanghai 200020, China
Tel: <86> (21) 6472-1001, Fax: <86> (21) 6415-2952

Renesas Technology Singapore Pte. Ltd.
1, Harbour Front Avenue, #06-10, Keppel Bay Tower, Singapore 098632
Tel: <65> 6213-0200, Fax: <65> 6278-8001

SH7709S Group
Hardware Manual

Renesas Electronics Corporation
1753, Shimonumabe, Nakahara-ku, Kawasaki-shi, Kanagawa 211-8668 Japan

REJ09B0081-0500O

X-ON Electronics

Largest Supplier of Electrical and Electronic Components

Click to view similar products for 32-bit Microcontrollers - MCU ***category:***

Click to view products by Renesas ***manufacturer:***

Other Similar products are found below :

MB91F575BHSPMC-GSE1 MB91F594BSPMC-GSE1 PIC32MX120F032B-50I/ML MB91F464AAPMC-GSE2 MB91F577BHSPMC-GSE1 SPC5604EEF2MLH MB91F528USCPMC-GSE2 MB91F248PFV-GE1 MB91F594BPMC-GSE1 MB91243PFV-GS-136E1 MB91F577BHSPMC1-GSE1 PIC32MM0032GPL020-E/ML PIC32MM0032GPL020-E/SS MEC1632X-AUE PIC32MM0016GPL020-E/ML PIC32MM0016GPL020-E/SS PIC32MM0016GPL028-E/SS PIC32MM0016GPL028-E/SO PIC32MM0016GPL028-E/ML PIC32MM0032GPL028-E/SS PIC32MM0032GPL028-E/SO PIC32MM0032GPL028-E/ML PIC32MM0032GPL036-E/MV PIC32MM0064GPL028-E/M6 PIC32MM0064GPL036-E/M2 PIC32MM0016GPL028-E/M6 PIC32MM0032GPL028-E/M6 MB91F526KSEPMC-GSE1 PIC32MM0064GPL028-E/SP PIC32MM0032GPL036-E/M2 TLE9872QTW40XUMA1 FT902L-T R5F564MLCDFB#31 R5F523E5ADFL#30 R5F524TAADFF#31 MCF51AC256ACPUE PIC32MM0064GPL028-I/ML PIC32MM0064GPL028-I/SP PIC32MM0064GPL028-I/SO PIC32MX120F032D-I/TL PIC32MX130F064D-I/ML PIC32MZ2064DAB169-I/HF PIC32MZ2064DAB288-I/4J ATUC256L4U-AUT R5F56318CDBG#U0 PIC32MX150F128C-I/TL PIC32MX170F256B-50IML PIC32MX130F064C-ITL PIC32MX230F064D-IML PIC32MM0032GPL028-I/ML